DTIC FILE COPY

2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A220 413

THESIS

DTIC ELECTE APR 12 1990 S E D

A KALMAN FILTER WITH SMOOTHING FOR HURRICANE TRACKING AND PREDICTION

by

Asim Mutaf

December 1989

Thesis Advisor Harold A. Titus

Approved for public release; distribution is unlimited.

90 04 11 123

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings
2a Security Classification Authority	3 Distribution/Availability of Report Approved for public release; distribution is unlimited.
2b Declassification Downgrading Schedule	
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School; 6b Office Symbol *(if applicable)* 62	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a Name of Funding Sponsoring Organization; 8b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*	10 Source of Funding Numbers: Program Element No; Project No; Task No; Work Unit Accession No

11 Title *(include security classification)* A KALMAN FILTER WITH SMOOTHING FOR HURRICANE TRACKING AND PREDICTION

12 Personal Author(s) Asim Mutaf

13a Type of Report	13b Time Covered	14 Date of Report *(year, month, day)*	15 Page Count
Master's Thesis	From To	December 1989	88

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	Kalman filter, Smoothing, Storm Tracking.

19 Abstract *(continue on reverse if necessary and identify by block number)*

The performance of a Kalman filter used to track a hurricane was substantially improved by implementing a fixed interval smoothing algorithm. This tracking routine was designed and implemented in a microcomputer program. Several tracking scenarios were simulated and analyzed. Actual storm tracks obtained from the Joint Typhoon Warning Center in Guam, Mariana Islands, were used for this research. The application of the Kalman tracker to a tropical storm's wind speed tracking was also investigated by using the best track data and observed data. Keywords: storm tracking; Computerized simulation. Theses, (EDC)

20 Distribution/Availability of Abstract ☒ unclassified unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual Harold A. Titus	22b Telephone *(include Area code)* (408) 646-2560	22c Office Symbol 62TS

DD FORM 1473,84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete — security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

A Kalman Filter With Smoothing for Hurricane Tracking and Prediction

by

Asim Mutaf
Lieutenant Junior Grade, Turkish Navy
B.S.E.E., Turkish Naval Academy, 1983

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE (EE)

from the

NAVAL POSTGRADUATE SCHOOL
December 1989

Author: ______________________
Asim Mutaf

Approved by: ______________________
Harold A. Titus, Thesis Advisor

____
Roberto Cristi, Second Reader

____
John P. Powers, Chairman,
Department of Electrical and Computer Engineering

ABSTRACT

The performance of a Kalman filter used to track a hurricane was substantially improved by implementing a fixed interval smoothing algorithm. This tracking routine was designed and implemented in a microcomputer program. Several tracking scenarios were simulated and analyzed. Actual storm tracks obtained from the Joint Typhoon Warning Center in Guam, Mariana Islands, were used for this research. The application of the Kalman tracker to a tropical storm's wind speed tracking was also investigated by using the best track data and observed data.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to Prof. Hal Titus for his guidance and constructive criticism which made this study a great learning experience. I also want to thank my wife Selime, whose love and support made this thesis "doable" and my son Nidal who kept me laughing the whole way.

I. INTRODUCTION

" Conceived over warm tropical oceans, born amid torrential thundershowers, and nurtured by water vapor drawn inward from far away, the mature tropical cyclone is an offspring of the atmosphere with both negative and positive consequences for life. Severe cyclones are among the most destructive of all natural disasters, capable of annihilating coastal towns and killing hundreds of thousands of people. On the positive though less dramatic side, they provide essential rainfall over much of lands they cross. It is difficult to convey to those who have never experienced a tropical cyclone the horror that great hurricanes can bring to ships at sea or people living near the coast. Tropical cyclones cause a variety of damage and the same tropical cyclone often affects several nations during its lifetime. They are called "Hurricanes" in the Atlantic and eastern Pacific" [Ref. 1]. Hurricanes were identified by female or male names like Pat and Tess. These storms will be discussed in this thesis. Tropical cyclones are also numbered sequentially according to their starting date. This numbering system is used with caution when referencing storms from other data bases.

This thesis attempted to improve the estimation of the hurricane's future course, speed, and position by using a Kalman filter with smoothing. This problem is similar to the ship tracking problem which is discussed in a previous thesis [Ref. 2]. The major difference between ship tracking and storm tracking problem is the measurement process which is given actual position coordinates (latitude and longitude) in the storm tracking problem. Therefore, the linearization required in the ship tracking problem is unnecessary in the storm tracking problem. The measurement noise varies with the type of the sensor (aircraft, satellite, and radar).

An accurate and reliable method of tracking and targeting is necessary. The current methods used to track a storm include the use of radar, aircraft, and satellite. However, the data may or may not be available when needed for a number of reasons. As an example, aircraft may not be available due to flight restrictions. A Kalman filter with a fixed interval smoothing algorithm can be used to track a storm. The smoothing algorithm is an off-line calculation that uses all measurements taken during a time interval $0 \leq k \leq M$ to improve the estimate. By having a more accurate assessment of what the storm has done in the past, we will be better able to predict ahead and estimate a storm's future course, speed, and position.

The estimation of the wind speed is as important as the storm position estimate. In an effort to estimate the possible damage a hurricane's sustained winds and storm surge could do to a coastal area, the Kalman filter and the smoother was used to estimate the wind speed and to categorize the hurricane. If the wind speed estimate is accurate, a hurricane is categorized correctly. This thesis attempts to estimate the hurricane's future wind speed. This will help to design a timely warning system.

II. PROBLEM STATEMENT

A. GENERAL

The storm-tracking scenario parallels the ship tracking problem in that both problems developed a position, course, and speed solution for a target with similar system dynamics. The tracking scenario used here involves two storms. The positions of the storms are given in x (longitude), and y (latitude) coordinates. This problem will be analyzed using state space methods. Given the longitude and latitude (the measurements) received by a radar, aircraft, or satellite, we are interested in estimating the location, course, and speed of the storm (the states of the plant). The state variables for this plant are x, $\dot{x}$, y, and $\dot{y}$.

B. SYSTEM MODEL

This system can be described by the state space equation

$$\underline{x}_{k+1} = \phi_k \underline{x}_k + w_k \tag{2.1}$$

where

$\underline{x}_k =$ state vector to be estimated,

$\phi_k =$ state transition matrix which describes how the states of the dynamic system are related, and

$w_k =$ random forcing function with a covariance matrix Q_k that is defined as

$$Q_k = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}, \tag{2.2}$$

The state vector is

$$\underline{x}_k = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} \tag{2.3}$$

and the system state equations are

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}_k + [w_k] \tag{2.4}$$

C. MEASUREMENT MODEL

The measurements are linearly related to the state variables, using the measurement equation

$$z_k = H_k x_k + v_k \tag{2.5}$$

Since the x and y position states are observed directly and given by latitude and longitude position coordinates, the measurement equation can be written as

$$\begin{bmatrix} z_x \\ z_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}_k + v_k \tag{2.6}$$

where the measurement noise v_k has a variance associated with the source of the measurement. In this thesis, mean deviation (nm) of satellite-derived tropical cyclone positions from best track positions (PCN values) were used in the calculation of the measurement noise covariance matrix for the satellite data. The measurement noise covariance matrix values and PCN values are shown in Table 1. The equation used in this calculation is

$$R_k = (Mean\ deviation)^2 \tag{2.7}$$

Table 1. THE MEASUREMENT NOISE COVARIANCE MATRIX VALUES FOR SATELLITE

PCN	Mean Deviation	R_k
1, or 2	16	256
3, or 4	30	900
5, or 6	40	1600

The measurement noise covariance matrix values were calculated by using the accuracy number for the aircraft and radar data. Equation (2.8) was used for aircraft data and Equation (2.9) was used for the radar data

$$R_k = \sqrt{((Navigational)^2 + (Meteorological)^2)} \qquad (2.8)$$

$$R_k = (Radar\ Accuracy)^2 \qquad (2.9)$$

where the radar accuracy numbers are shown in Table 2

Table 2. THE MEASUREMENT NOISE COVARIANCE MATRIX VALUES FOR RADAR

Accuracy Number	Radar Accuracy	R_k
1, 4, or G	10	100
2, 5, or F	15	225
3, 6, or P	25	625
7, or Blank	30	900

D. KALMAN FILTER

Basically, the Kalman filter takes an a priori estimate of the states, projects it ahead in time to some predicted estimate, and then calculates a gain vector based on the error covariance of these estimates. The error between the observed measurements and the predicted measurements of the corresponding state estimates is multiplied by the gain vector and the result is added to the predicted state estimates to give the best estimate of the true states based on optimal combinations of a priori estimates and current measurements.

The Kalman filter is the proper algorithm to be used when both the system model and the measurement model are linear functions of the state variables. The basic operation of the filter is a relatively straightforward recursive process. The equations used in the Kalman filter [Ref. 3] are

$$\underline{x}_{k+1} = \phi_k \underline{x}_k + \Gamma_k \underline{w}_k \qquad (2.10)$$

$$\underline{z}_k = H_k \underline{x}_k + \underline{v}_k \qquad (2.11)$$

$$\hat{\underline{x}}_{(k|k-1)} = \phi_k \hat{\underline{x}}_{(k|k)} \qquad (2.12)$$

$$P_{(k|k-1)} = \phi_k P_{(k|k)} \phi_k^T + Q_k \qquad (2.13)$$

$$G_k = P_{(k|k-1)} H_k^T (H_k P_{(k|k-1)} H_k^T + R_k)^{-1} \qquad (2.14)$$

$$\hat{x}_{(k|k)} = \hat{x}_{(k|k-1)} + G_k(z_k - H_k\hat{x}_{(k|k-1)}) \tag{2.15}$$

$$P_{(k|k)} = (I - G_k H_k)P_{(k|k-1)} \tag{2.16}$$

where

$\hat{x}_{(k|k-1)}$ = projected ahead state estimate,

$P_{(k|k-1)}$ = projected ahead state error covariance matrix,

G_k = Kalman gain matrix,

R_k = state measurement noise covariance matrix, and

H_k = linearized measurement matrix.

The Kalman gain matrix serves to minimize the mean square estimation error and is an indication of how much weight will be placed on the current observation. A large gain, indicating a large error covarince, will place more weight on the current observation as the filter tries to correct the states. The gain matrix is proportional to the variance of the uncertainty in the estimate and inversely proportional to the variance of the measurement noise. It can be expressed as

$$G_k = P_{(k|k-1)} H_k^T R_k^{-1} \tag{2.17}$$

An initial velocity estimate is taken to be zero since there is no velocity information at the beginning. The initial state estimates carry with them some error and it is this error, or rather an estimate of this error, that is used to construct the initial error covariance matrix. The initial position error was estimated to be 10 nautical miles in the $x\,y$ direction and the initial velocity was estimated to be 0.158 nautical miles per minute. The error was assumed to be zero mean and uncorrelated. With these approximations, the initial error covariance matrix is given by

$$P_{(0|-1)} = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 0.025 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0.025 \end{bmatrix} \tag{2.18}$$

E. SMOOTHING ALGORITHM

Smoothing is a procedure that uses all of the state estimates produced by an estimator and attempts to improve the accuracy of these estimates by using the negative

time dynamics to produce the smoothed estimate. The estimator used here is the Kalman filter. The basic idea behind smoothing is that, for a time interval from 0 to K ($K > k$), an estimate at time k based on all previous estimates up to time K, ($\hat{x}_{(k|K)}$), will be more accurate than an estimate based only on the estimates up to time k, ($\hat{x}_{(k|k)}$). " It is a non-real time operation where the available data are processed to obtain an estimate $\hat{x}_{(k|K)}$ for some past value of k " [Ref. 4].

Smothing algorithms were categorized into three groups by Meditch [Ref. 5];

Fixed Point Smoothing smooths the estimate $\hat{x}_{(k|K)}$ at a fixed point k as K increases.

Fixed Lag Smoothing smooths the estinate $\hat{x}(K - N \mid K)$ at a fixed delay N as K increases.

Fixed Interval Smoothing smooths the estimate $\hat{x}_{(k|K)}$ over the time interval from 0 to K where K is fixed and k varies from 0 to K.

A fixed-interval smoothing algorithm was used in this thesis. This smoothing routine provides the optimal state estimate at each time k over a fixed interval from 0 to K. The equations used in the smoothing algorithm [Ref. 5] are

$$A_k = P_{(k|k)} \Phi^T P^{-1}_{(k+1|k)} \tag{2.19}$$

$$\hat{x}_{(k|N)} = \hat{x}_{(k|k)} + A_k(\hat{x}_{(k+1|N)} - \hat{x}(k+1 \mid k)) \tag{2.20}$$

$$P_{(k|N)} = P_{(k|k)} + A_k(P_{(k+1|N)} - P_{(k+1|k)})A_k^T \tag{2.21}$$

where

A_k = smoothing filter gain matrix,

$\hat{x}_{(k|N)}$ = smoothed state estimate a time k based on N observations, and

$P_{(k|N)}$ = smoothed state error covariance matrix.

At the beginning of the smoothing, the last filtered estimate becomes the initial smoothed estimate. The index k is decremented by one for each pass during the smoothing algorithm with the starting value of k equal to the number of data points to be smoothed, minus one ($N - 1$). Consequently, the tracking program makes ($N - 1$) passes through the smoothing algorithm.

III. STORM TRACKING

A. GENERAL

The Kalman filter program STORM.FOR was used in computer simulations. This program was originally written for a ship tracking problem and was modified to use on storm tracking problem. The graphing routines of the MATLAB were used to generate the graphs. A complete listing of the program is included in Appendix A. Typhoon Tess and Typhoon Pat were used for simulations. The storm tracks used were obtained from data collected at the Joint Typhoon Warning Center located in Guam. Each storm is given a separate deck name. Tropical cyclones are numbered sequentially according to their starting date by the JTWC. There are four types of data:

Best Track -This file is the 6-hourly storm positions based on a post storm, subjectively smoothed path.

Forecasts -This data contains the real time storm positions, objective forecasts, and the official forecast. Each date-time group may contain one, two, or all three types of data.

Forecast Errors -Eight different errors were computed for each of the objective and official forecasts.

Fixes -Tropical cyclone fixes (observations) from four different platforms are contained in the data base.

The position coordinates were obtained using aircraft, satellite, and radar. The data obtained included: raw data (observations); best track data; and 12, 24, and 48 hour predictions. The raw data was processed just as if it was real-time observation of the hurricane. The first storm, Pat, originated east of Taiwan in the western Pacific on 24 August 1985. The warning period for this storm was six days. The storm traveled 1337 nm. The maximum speed of the wind was over 107 kt and the minimum sea level pressure was 1002 mb. The Typhoon Pat caused significant damage in southwestern and northeastern Japan; primarly on the islands of Kyushu and Hokkaido. Kyushu was hit the hardest with wind gusts of 107 kt. A total of 23 people were reported killed with over 180 people injured. An estimated 3000 homes were damaged. Pat also severely disrupted transportation by land, sea, and air.

The second storm track analyzed was that of Typhoon Tess which originated southeast of Guam on 30 August 1985. The warning period for this storm was seven days. The storm traveled 1470 nm with maximum wind speeds of over 90 kt. The storm

brought needed rain to the Philippines during a spell of drier than normal weather. The storm also brought death and destruction. Considerable flooding and crop damage occurred over southern China as Tess moved inland [Ref. 6]. The observed track of Typhoon Pat and Typhoon Tess are shown in Figure 1 and Figure 2, respectively.

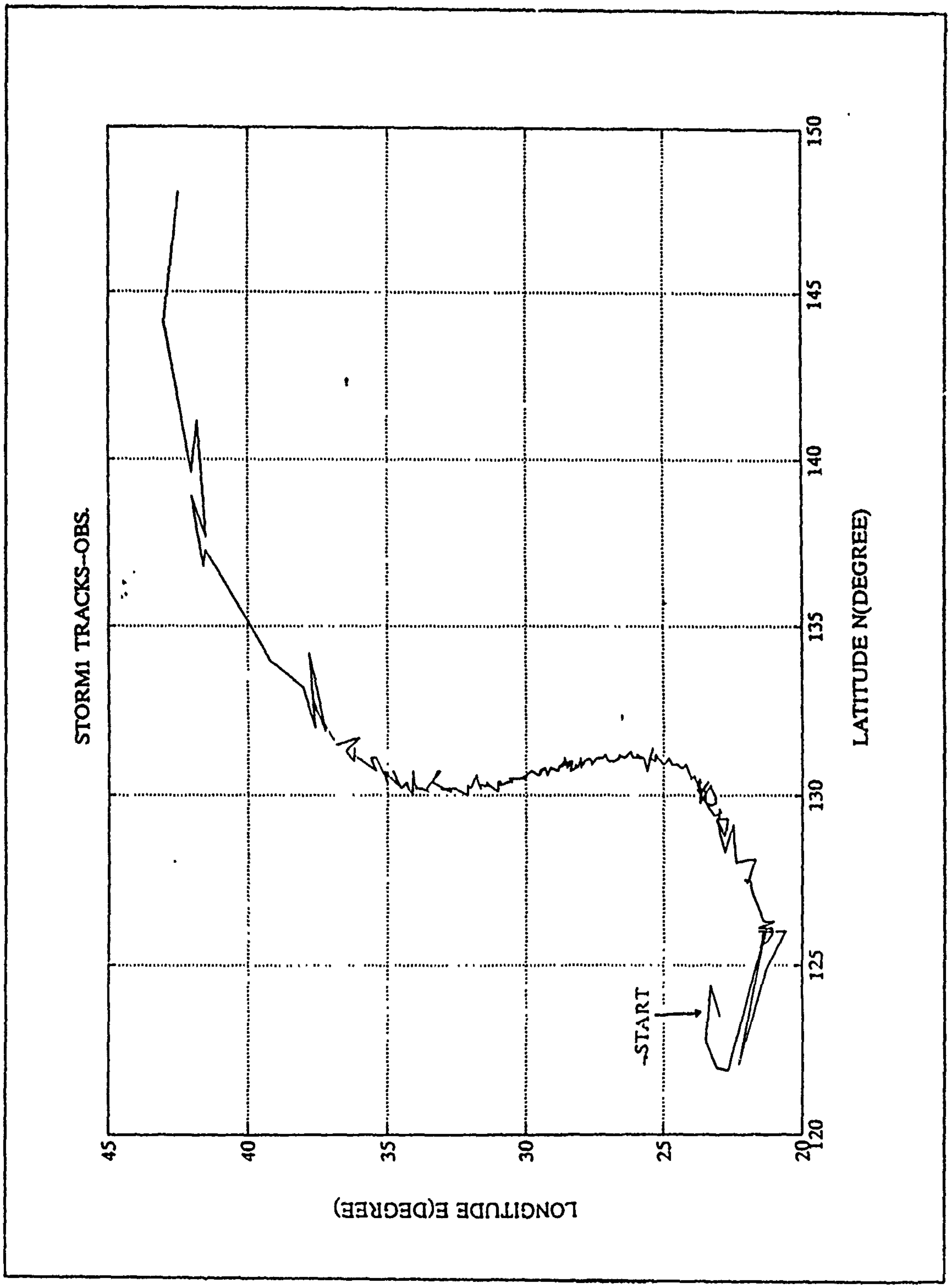

Figure 1. The observed track of Typhoon Pat [Ref. 6]

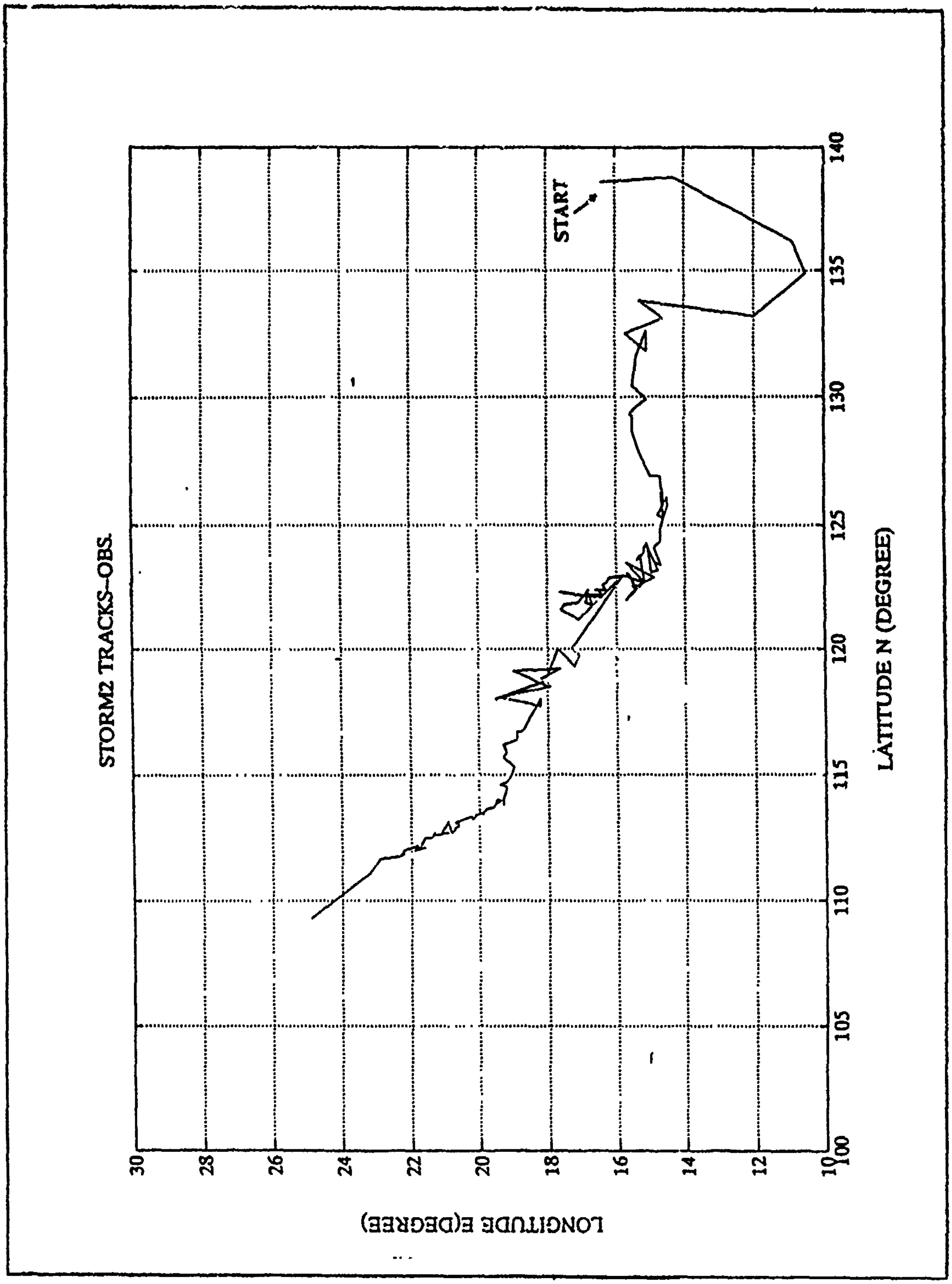

Figure 2. The observed track of Typhoon Tess [Ref. 6]

B. COMPUTER SIMULATIONS

1. Typhoon Pat

The best-track of Typhoon Pat is shown in Figure 3. The best track positions are in 6-hourly increments. The first tracking data point corresponds to the day-time group 08270000Z. Figure 4 shows the Kalman filter position estimates and Figure 5 shows the smoothed position estimates. Figure 6 was constructed using the filtered and smoothed position estimates. In general, the smoother does improve the track accuracy. In the area of the track where the true positions do vary, the smoother tracking error is zero. Specifically, this area occurs between 23° N, 124° E, and 38° N, 133° E. This area can be seen easily in Figure 7. This figure was constructed by using the tracking error of the filter and smoother. The average tracking errors for this storm are $\pm$ 4 nautical miles for the filter and $\pm$ 2 nautical miles for the smoother estimates.

2. Typhoon Tess

The performance of the smoother on the track of Typhoon Tess was similar to that of Typhoon Pat. Figure 8 shows Typhoon Tess best track. Typhoon Tess best track data are also in 6-hourly increments. The filter and smoother tracking results are shown in Figure 9 and Figure 10, respectively. Figure 11 shows the track results obtained with the Kalman filter and smoothing algorithm. The smoother shows some improvement near 17.5° N, 120°E and 15.2° N, 130°E. The filter average tracking error increased slightly, to about $\pm$ 5 nm, but the smoother average tracking error jump to about $\pm$ 5 nm. This is because the smoother gives 30 nautical miles tracking error near 18.8° N, 116°E due to large change on the direction of Typhoon Tess. Figure 12 shows the tracking errors of the filter and smoother. It is observed that the smoother was much less sensitive to the large course changes than the Kalman filter. It is, therefore, reasonable to assume that similar results could be expected from the smoother for a large course change more than 90° . However, the smoother's estimates are quite good over the entire trajectory and the estimates closely follow the course changes as in Typhoon Pat.

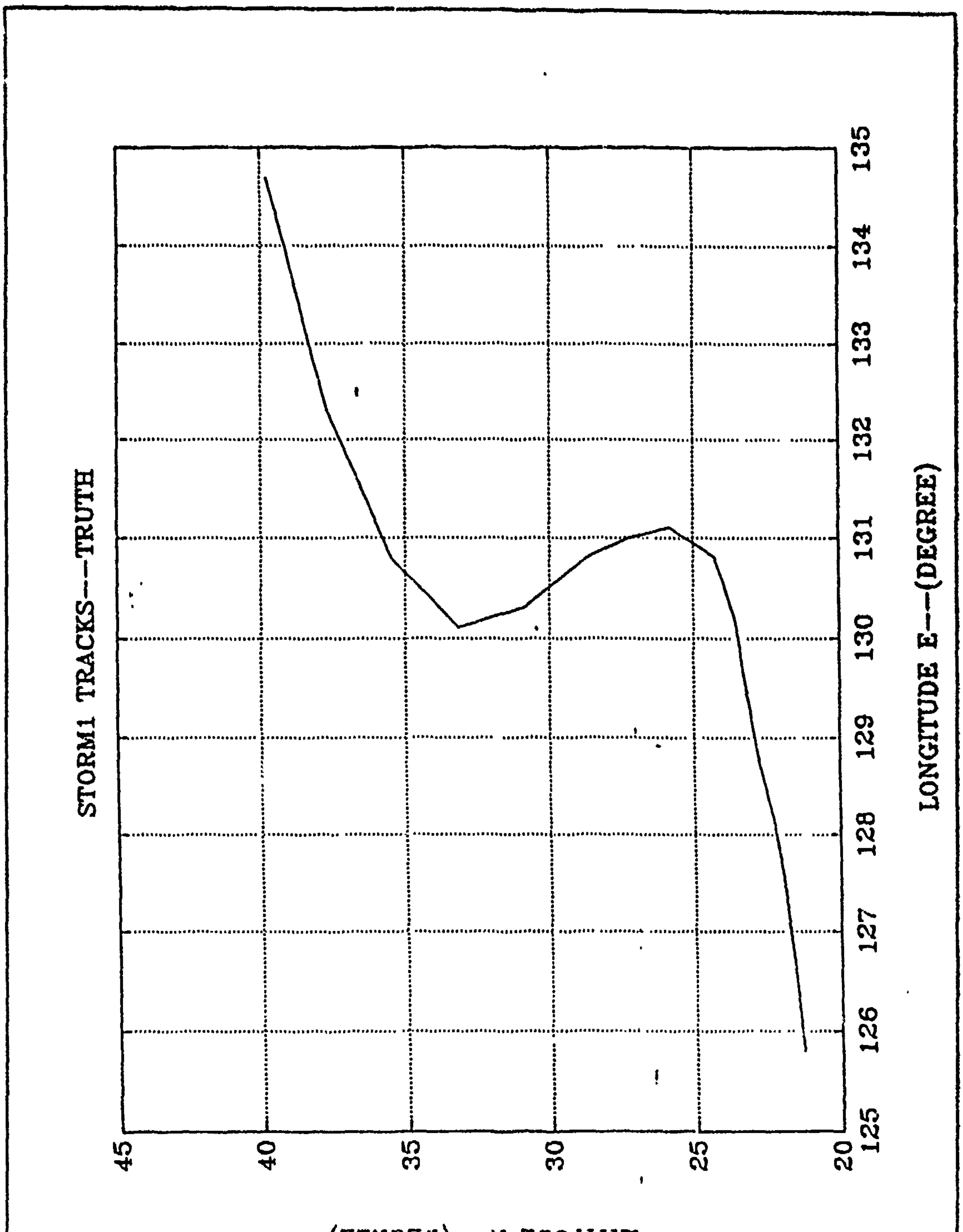

Figure 3. The Best Track of Typhoon Pat

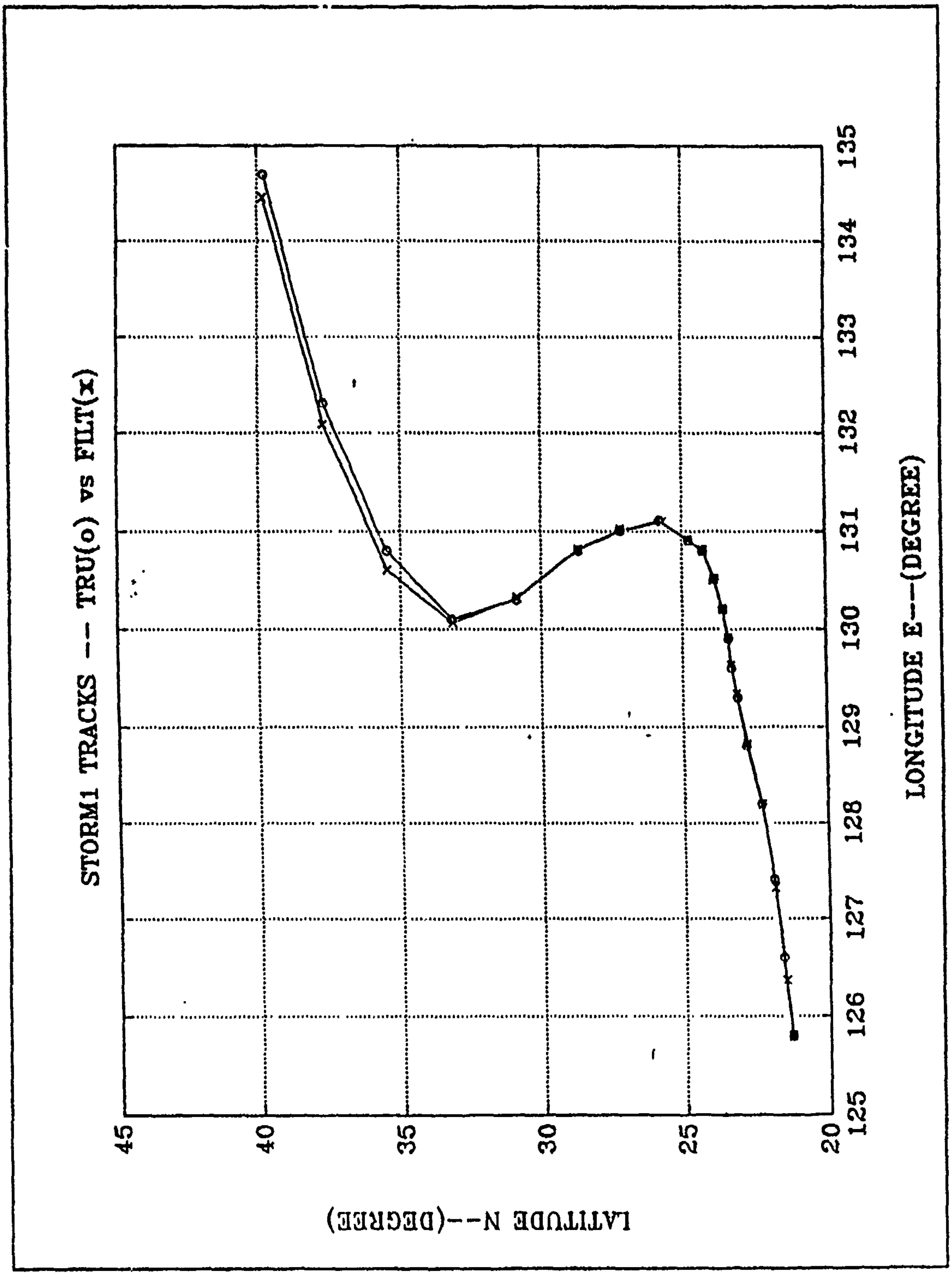

Figure 4. Filtered Track of Typhoon Pat

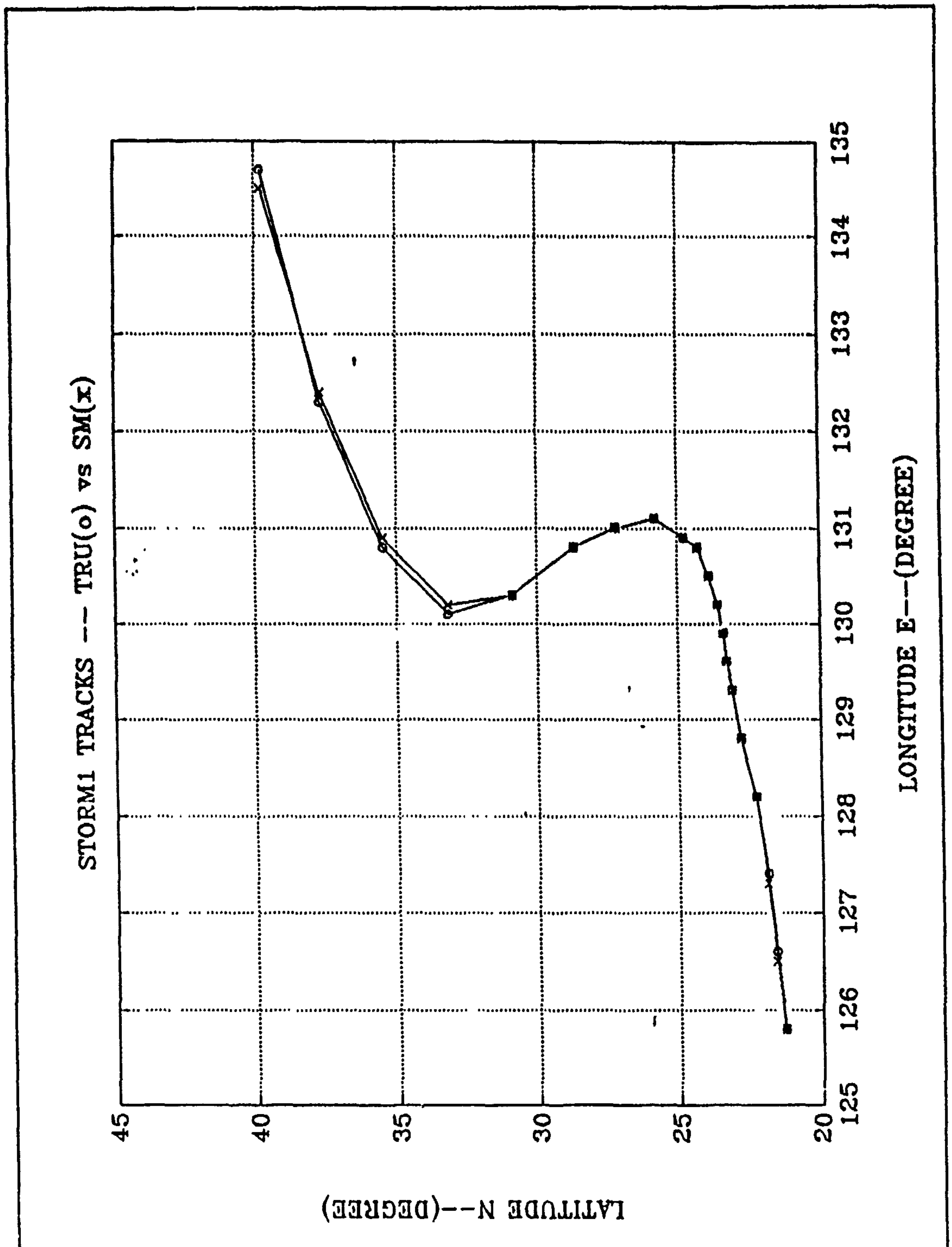

Figure 5. Smoothed Track of Typhoon Pat

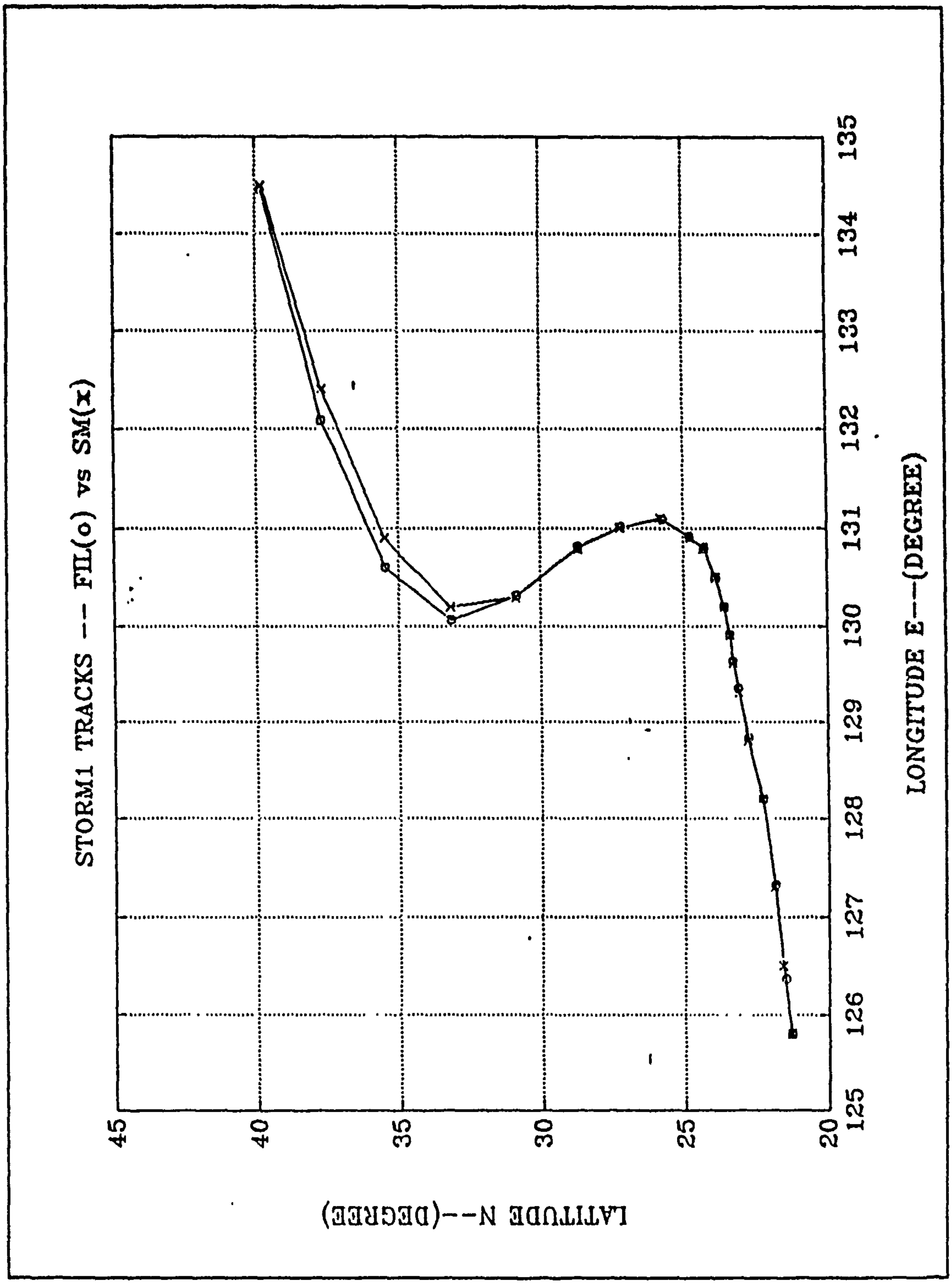

Figure 6.   Filtered and Smoothed Track of Typhoon Pat

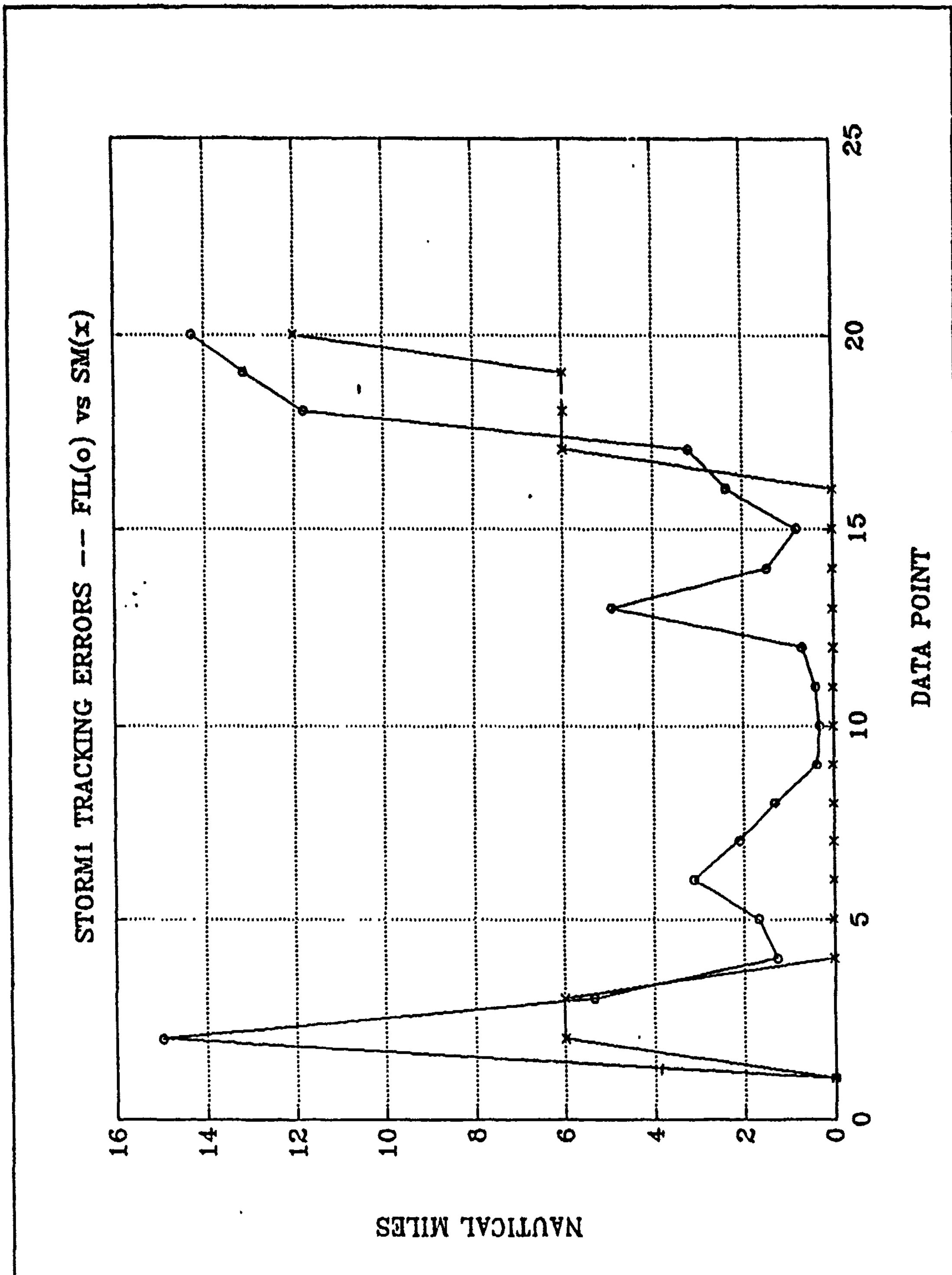

Figure 7. Tracking Errors of the Filter and Smoother for typhoon Pat

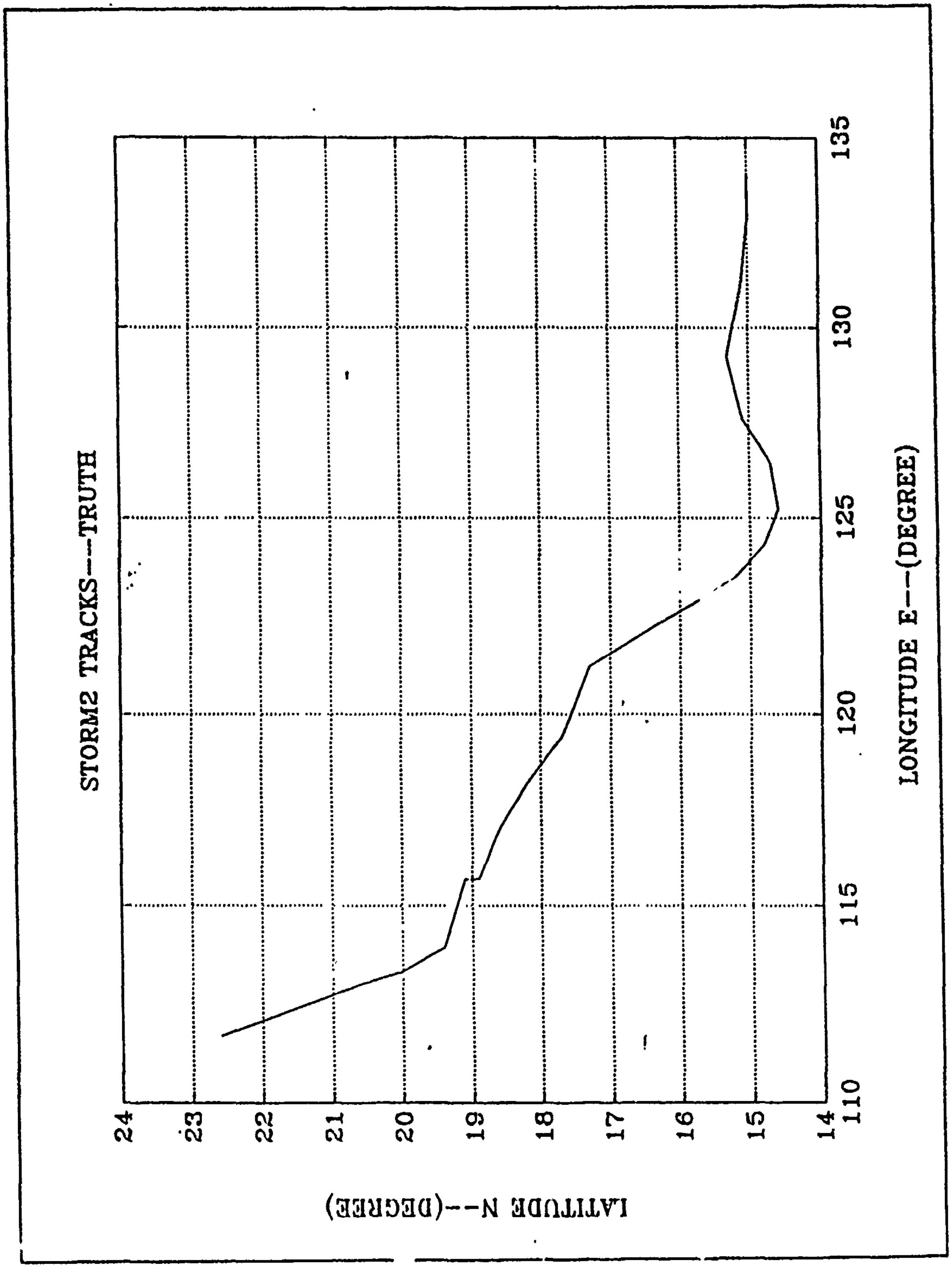

Figure 8. The Best Track of Typhoon Tess

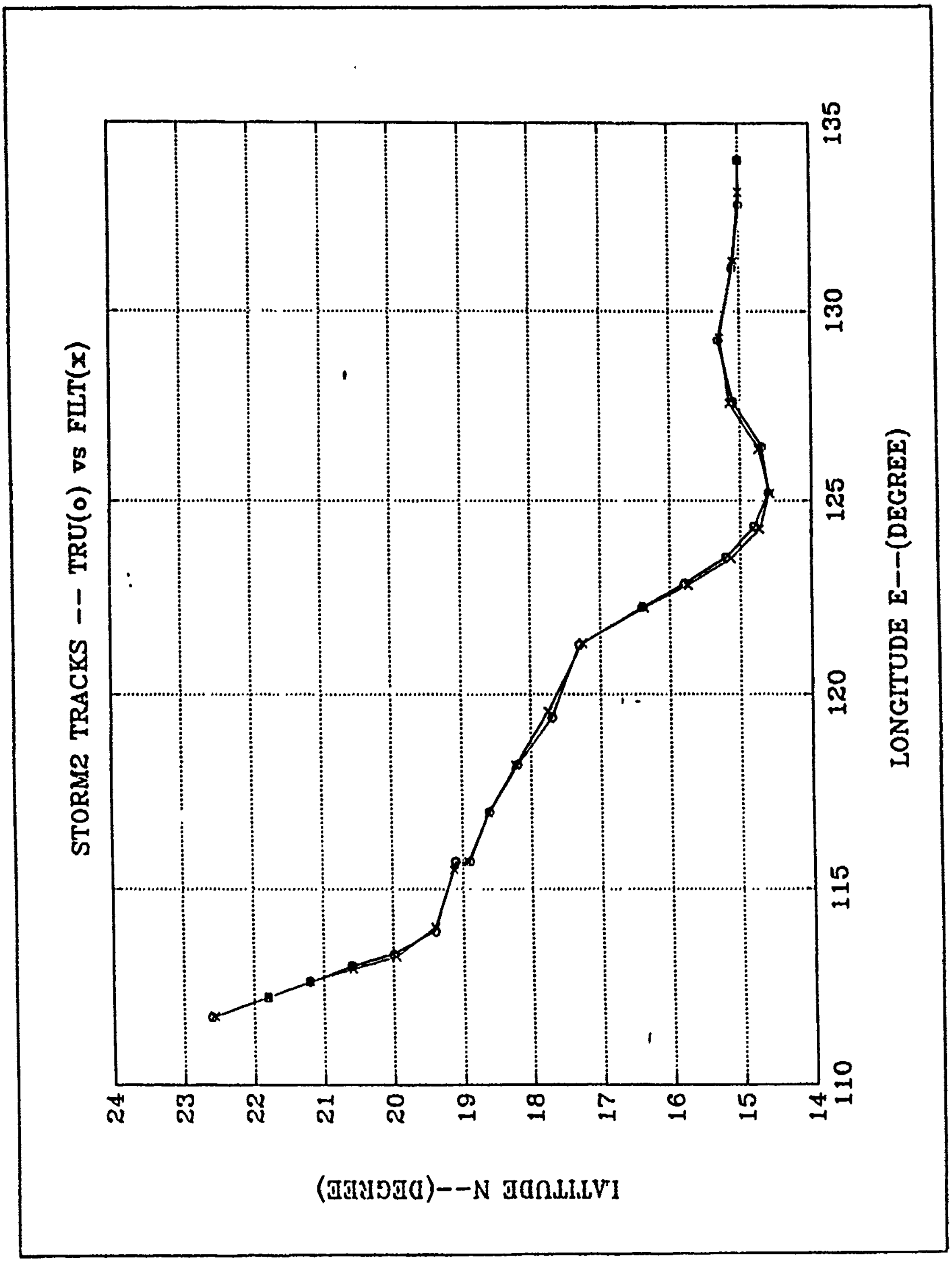

Figure 9. Filtered Track of Typhoon Tess

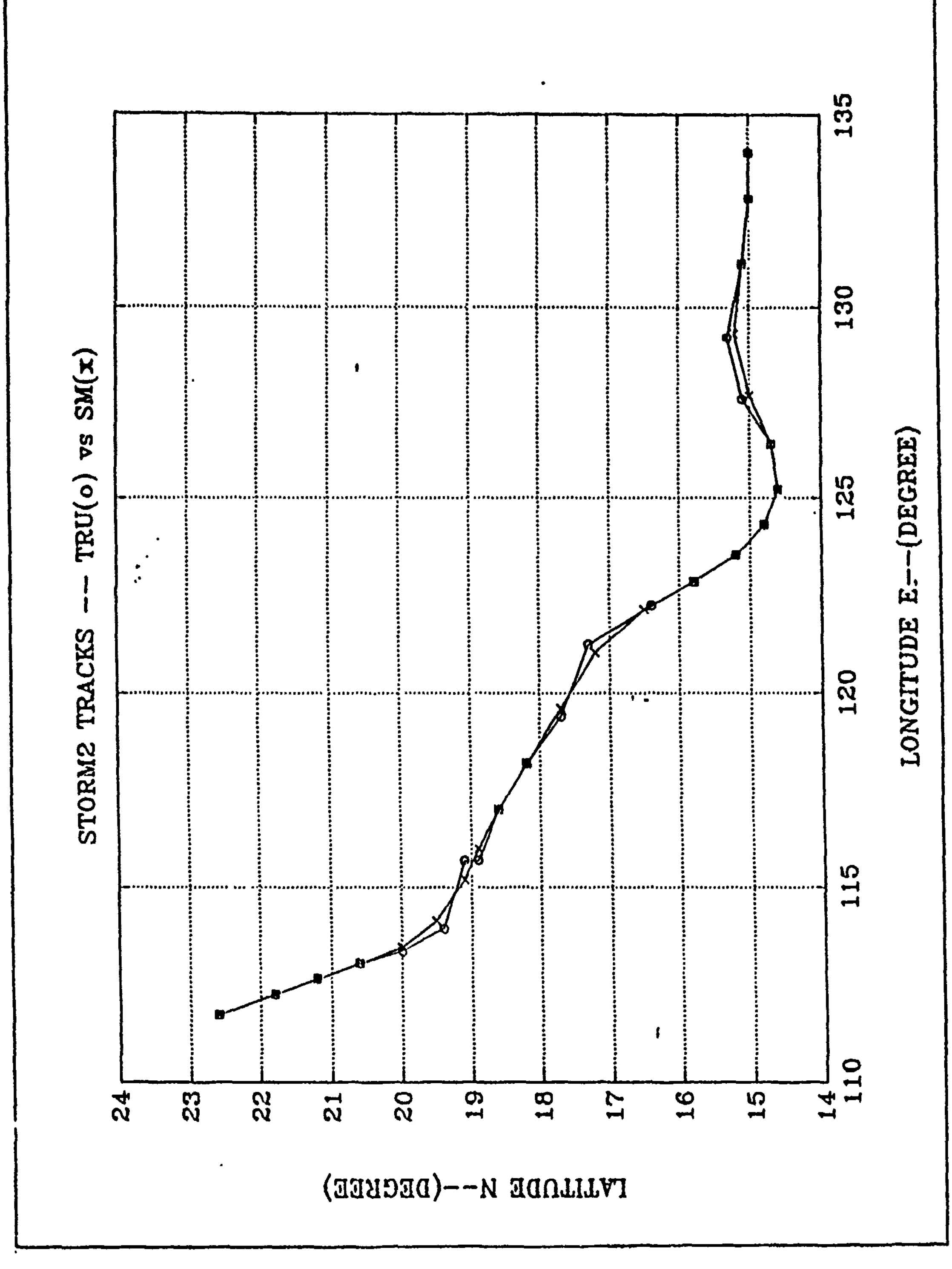

Figure 10. Smoothed Track of Typhoon Tess

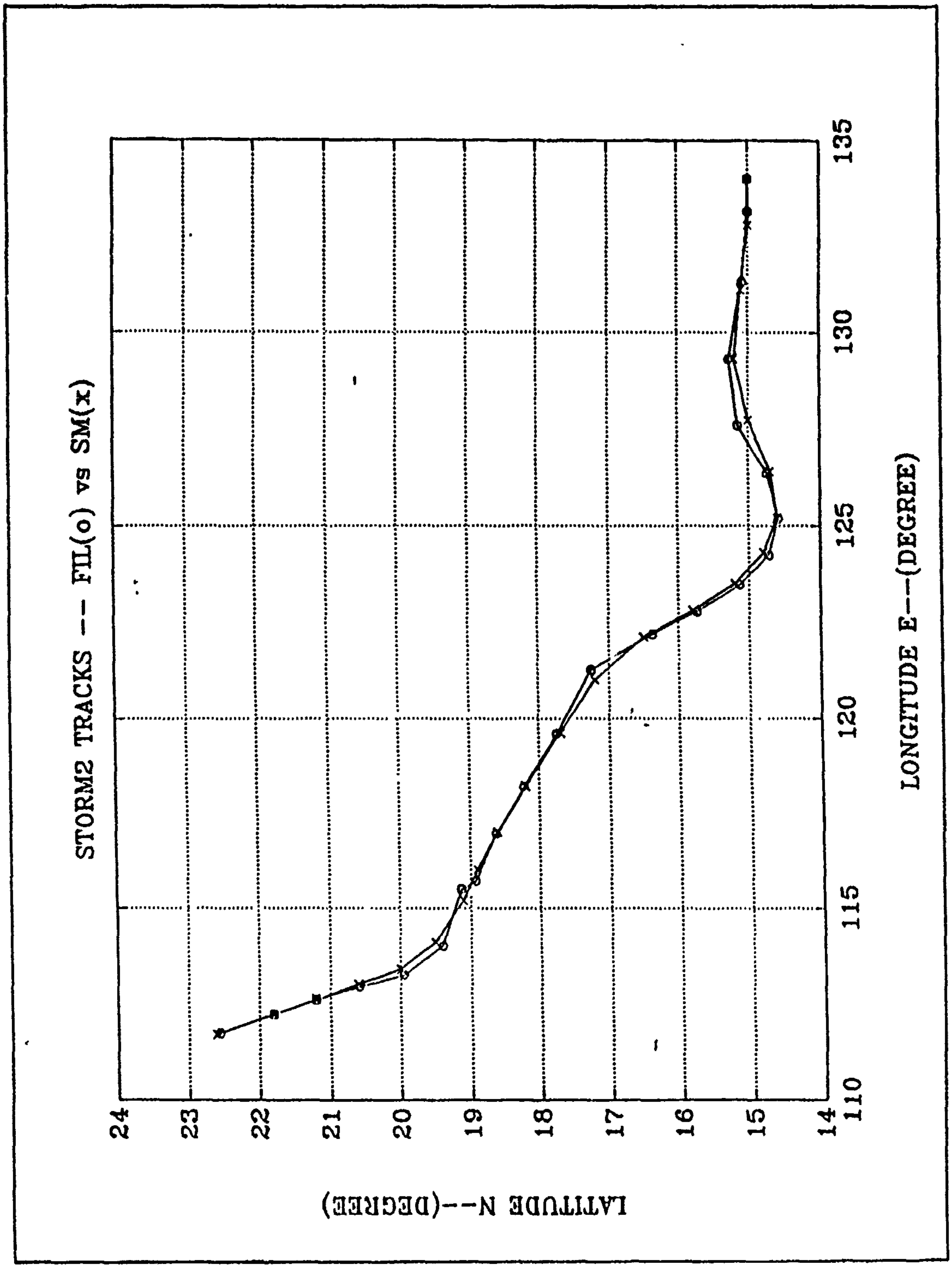

Figure 11. Filtered and Smoothed Track of Typhoon Tess

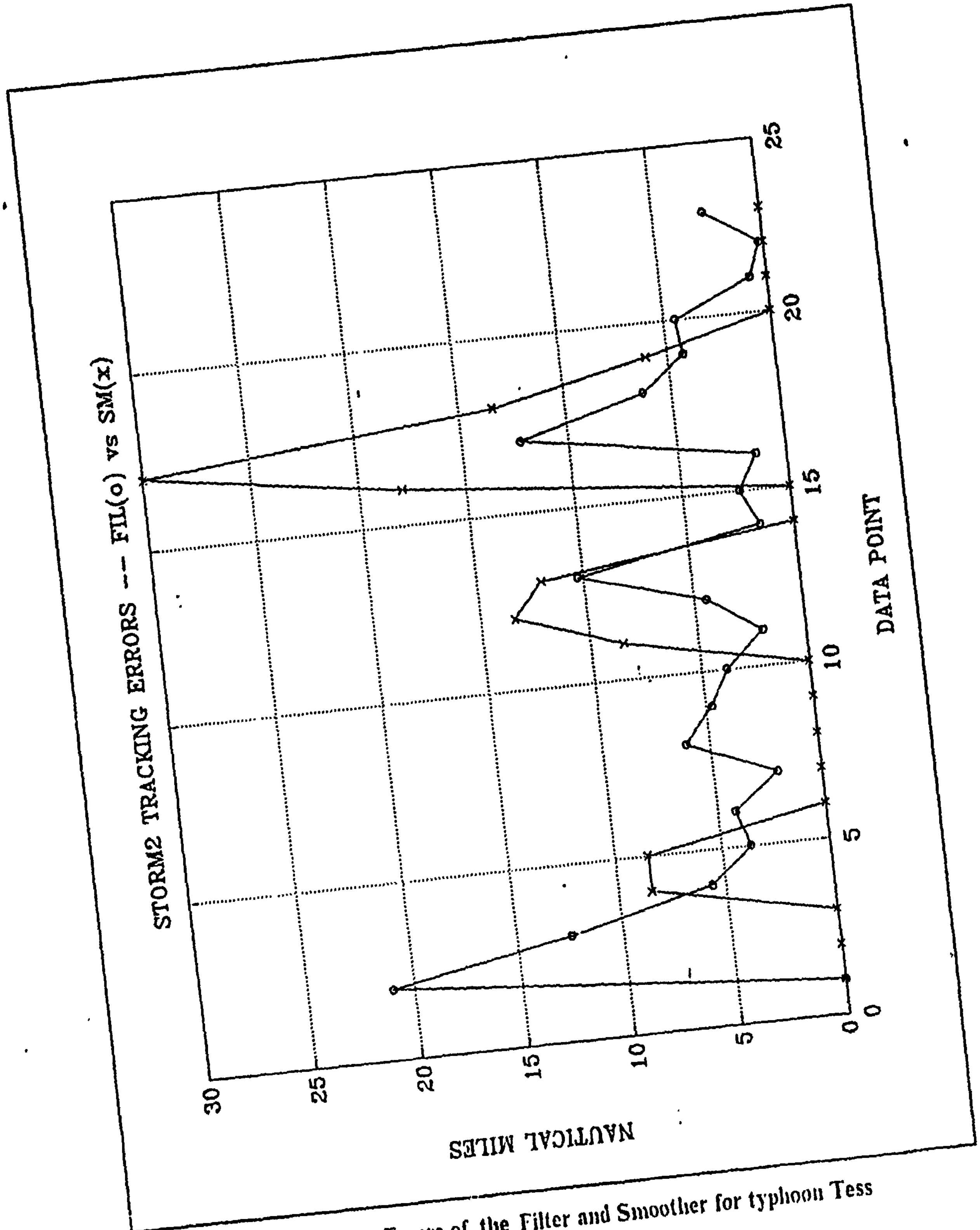

Figure 12. Tracking Errors of the Filter and Smoother for typhoon Tess

IV. STORM WIND TRACKING

A. GENERAL

" In an effort to estimate the possible damage a hurricane's sustained winds and storm surge could do to a coastal area, the Saffir-Simpson damage-potential scale was developed. The scale numbers are based on actual conditions at some time during the life of the storm " [Ref. 7]. Table 3 shows these categories.

Table 3. SAFFIR-SIMPSON HURRICANE DAMAGE-POTENTIAL SCALE

Scale Number	Wind speed(knots)	Damage
1	64-82	Damage mainly to trees, shrubbery, and unanchored mobile homes.
2	83-95	Some trees blown down; major damage to exposed mobile homes; some damage to roofs of buildings.
3	96-113	Foliage removed from trees; large trees blown down; mobile homes destroyed; some structural damage to small buildings.
4	114-135	All signs blown down; extensive damage to roofs, windows, and doors; complete destruction of mobile homes; flooding inland as far.
5	> 135	Severe damage to windows and doors; extensive damage to roofs of homes and industrial buildings; small buildings overturned and blown away; major damage to lower floors of all structures less than 4.5 m above sea level within 500 m of shore.

The storm wind tracking scenario parallels the storm tracking problem. The tracking scenario used here involves two storms. This problem will be analyzed using state space methods. Given the tropical cyclone intensity values the observed speed of the storm wind will be estimated by using the Kalman filter and smoother. Table 4 shows the relationship between intensity and wind speed. The wind speed was used directly as a measurement for the best track data of the storm. The state variables for this plant are $\dot{w}$, and $\ddot{w}$.

The system can be described by the state space equation

$$\underline{w}_{k+1} = \phi_k \underline{w}_k + f_k \tag{4.1}$$

where

$\underline{w}_k$ = state vector to be estimated,

ϕ_k = state transition matrix which describes how the states of the dynamic system are related, and

f_k = random forcing function with a covariance matrix Q_k that is defined as

$$Q_k = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} \\ \frac{T^3}{2} & T^2 \end{bmatrix} E[(f_k)^2] \tag{4.2}$$

The state vector is

$$\underline{w}_k = \begin{bmatrix} \dot{w} \\ \ddot{w} \end{bmatrix} \tag{4.3}$$

and the system state equations are

$$\begin{bmatrix} \dot{w} \\ \ddot{w} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{w} \\ \ddot{w} \end{bmatrix}_k + [f_k] \tag{4.4}$$

The measurements are linearly related to the state variables. Using the measurement equation

$$\underline{z}_k = H_k \underline{w}_k + \underline{v}_k \tag{4.5}$$

The measurement equation can be written as

$$z_{k+1} = [\,1\ 0\,] \begin{bmatrix} \dot{w} \\ \ddot{w} \end{bmatrix}_k + \underline{v}_k \tag{4.6}$$

where the measurement noise $\underline{v}_k$ has a variance associated with the source of the measurement. The measurement noise covariance matrix values are calculated in the same manner as in storm position tracking problem by using Equations (2.8) and (2.9) for the aircraft and radar data.

The initial error covariance matrix used in the wind speed tracking is

$$P_{(0|-1)} = \begin{bmatrix} 1000000 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 1000000 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix} \tag{4.7}$$

Table 4. MAXIMUM SUSTAINED WIND SPEED AS A FUNCTION OF FORECAST INTENSITY NUMBER

Intensity	Wind speed(nm/h)
00	25
05	25
10	25
15	25
20	30
25	35
30	45
35	55
40	65
45	77
50	90
55	102
60	115
65	127
70	140
75	155
80	170

B. COMPUTER SIMULATIONS

1. The Best Track Data

a. Typhoon Pat

Using the best track data wind speed values as the measurements, future wind speed values were estimated by the filter and the smoother. There is an initial track error due to the error in the initial state estimates. When the wind speed increases at 24 hours, the tracking error decreases and becomes zero for the fifth data as the filter gains the wind track. However, it increases after 90 hours when the wind speed decreases very fast and it returns to zero two data points later as the filter regains the wind

track. Figure 14 shows the filter tracking accuracy. The smoother is not as accurate as in the position estimate due to the large change in wind speed, but these errors remain in the acceptable ranges. The smoother track is shown in Figure 15. The average tracking errors are $\pm$ 0.5 mph for the filter and $\pm$ 1.1 mph for the smoother. Best track data represents the weather service's estimate of truth [Ref. 6]. Figure 16 compares the forward time estimate (filter, FIL(o)) with the forward and negative time estimate (smoother, SM(x)) for Typhoon Pat. Figure 17 denotes the error in these estimates.

b. Typhoon Tess

The tracking results for this storm are shown in Figures 18-22. From Figure 19 and 20, we can see how the Kalman filter and the fixed interval smoothing improve the overall track estimate. During the overall track estimate, two large filter tracking errors are detected. This is shown in Figure 19. In both instances the smoother also gives large tracking errors. Figure 20 shows the smoother estimates. At other times, however, the filtered and smoothed estimate are accurate. Figure 21 is the comparison of the filter and smoother estimates. The filter average tracking error is $\pm$ 1.5 mph and the smoother average tracking error is $\pm$ 2.0 mph. Figure 22 shows the tracking errors of the filter and smoother estimates.

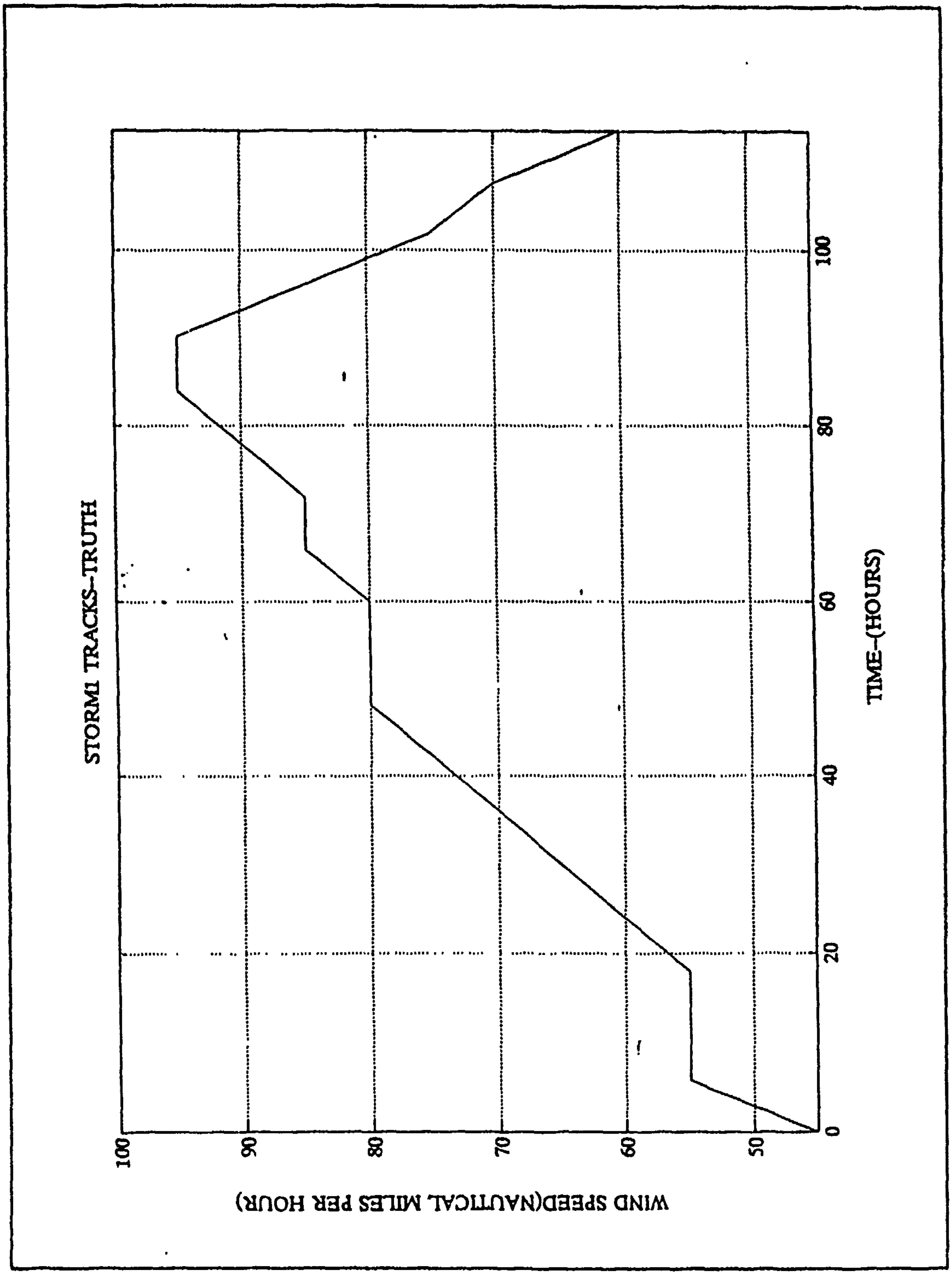

Figure 13. The Best Track Wind Speed of Typhoon Pat [Ref. 6]

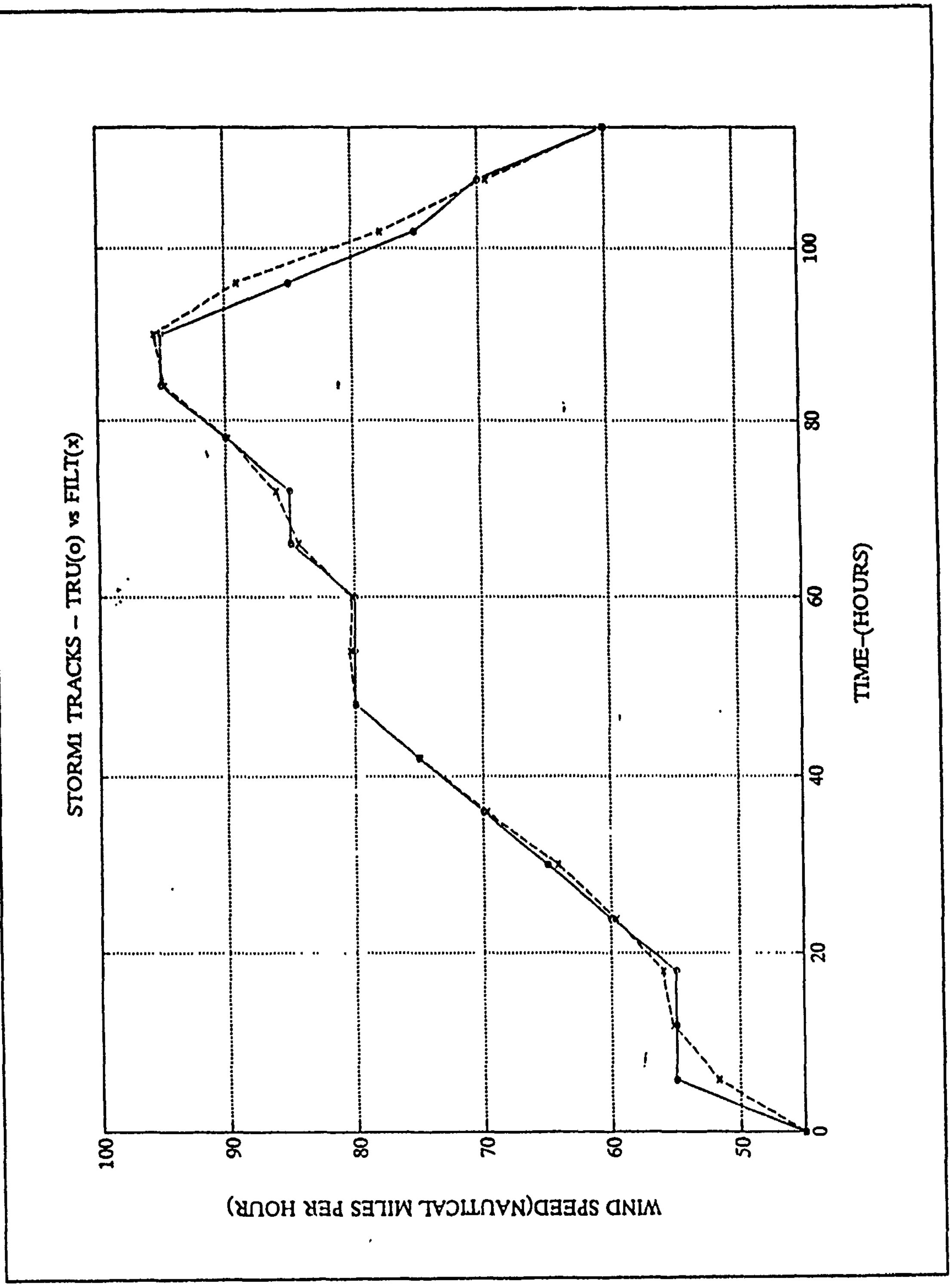

Figure 14. Filtered Track of Typhoon Pat's Best Track Wind Speed

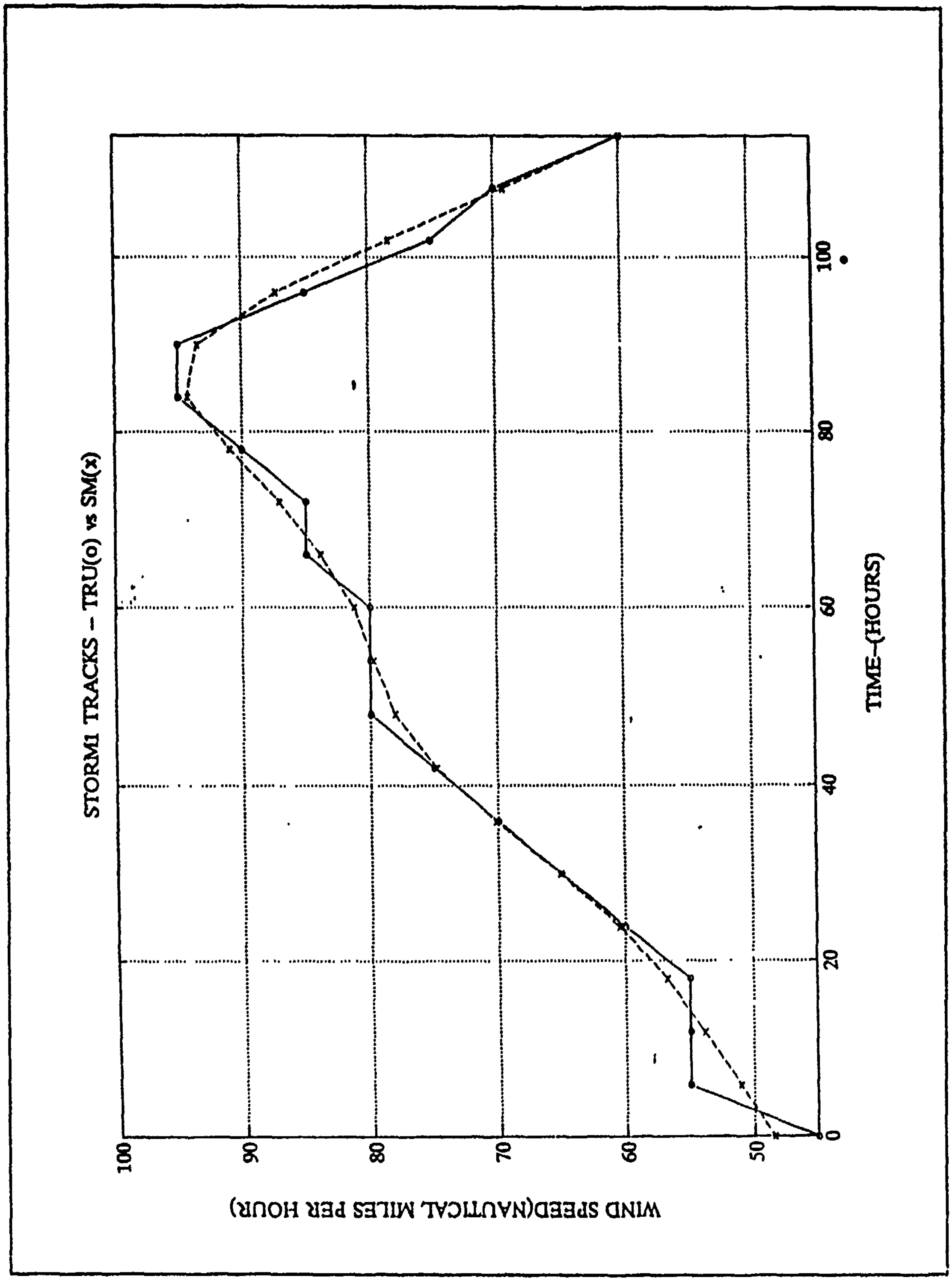

Figure 15. Smoothed Track of Typhoon Pat's Best Track Wind Speed

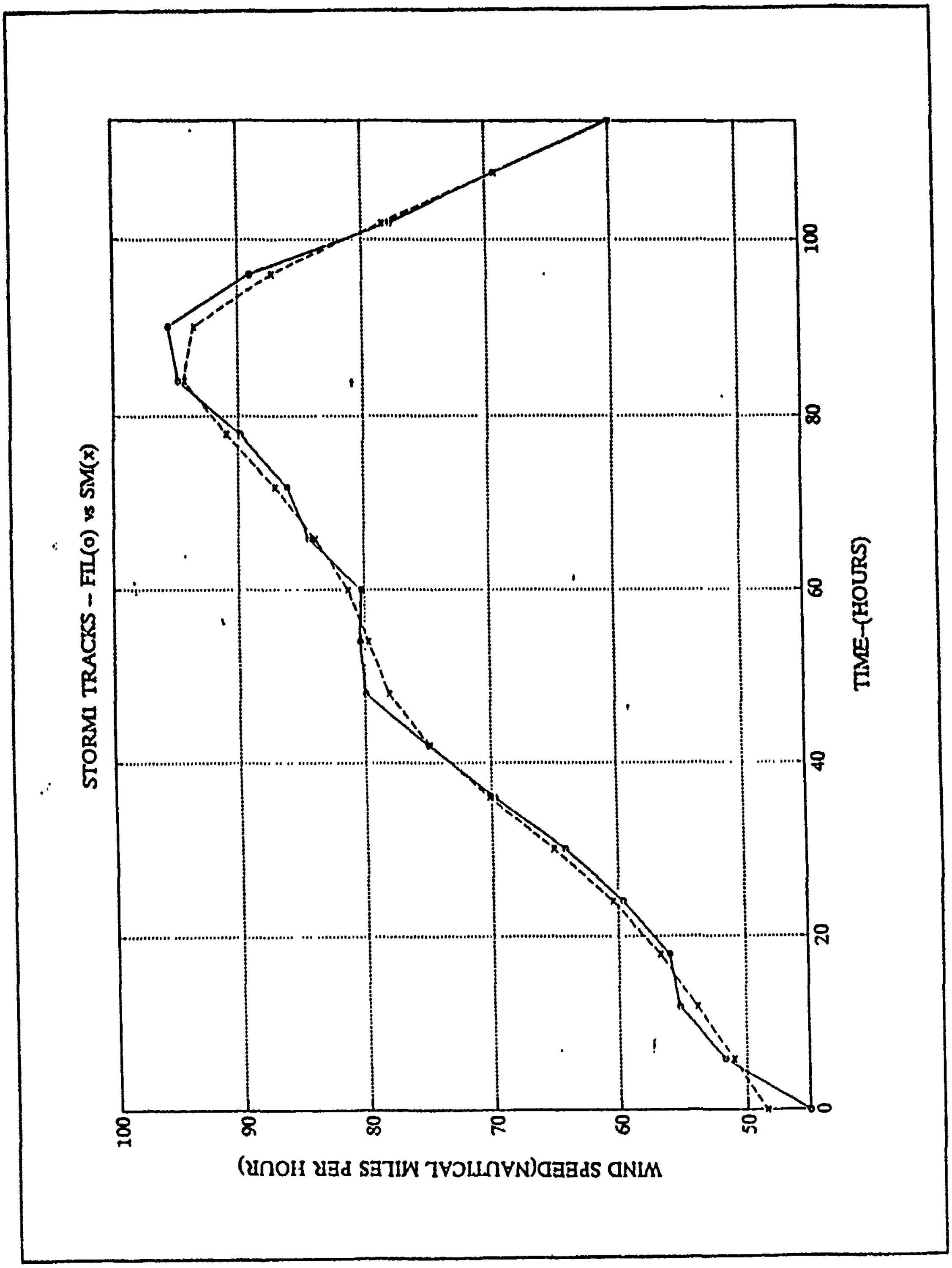

Figure 16. Filtered and Smoothed Track of Typhoon Pat's Best Track Wind Speed

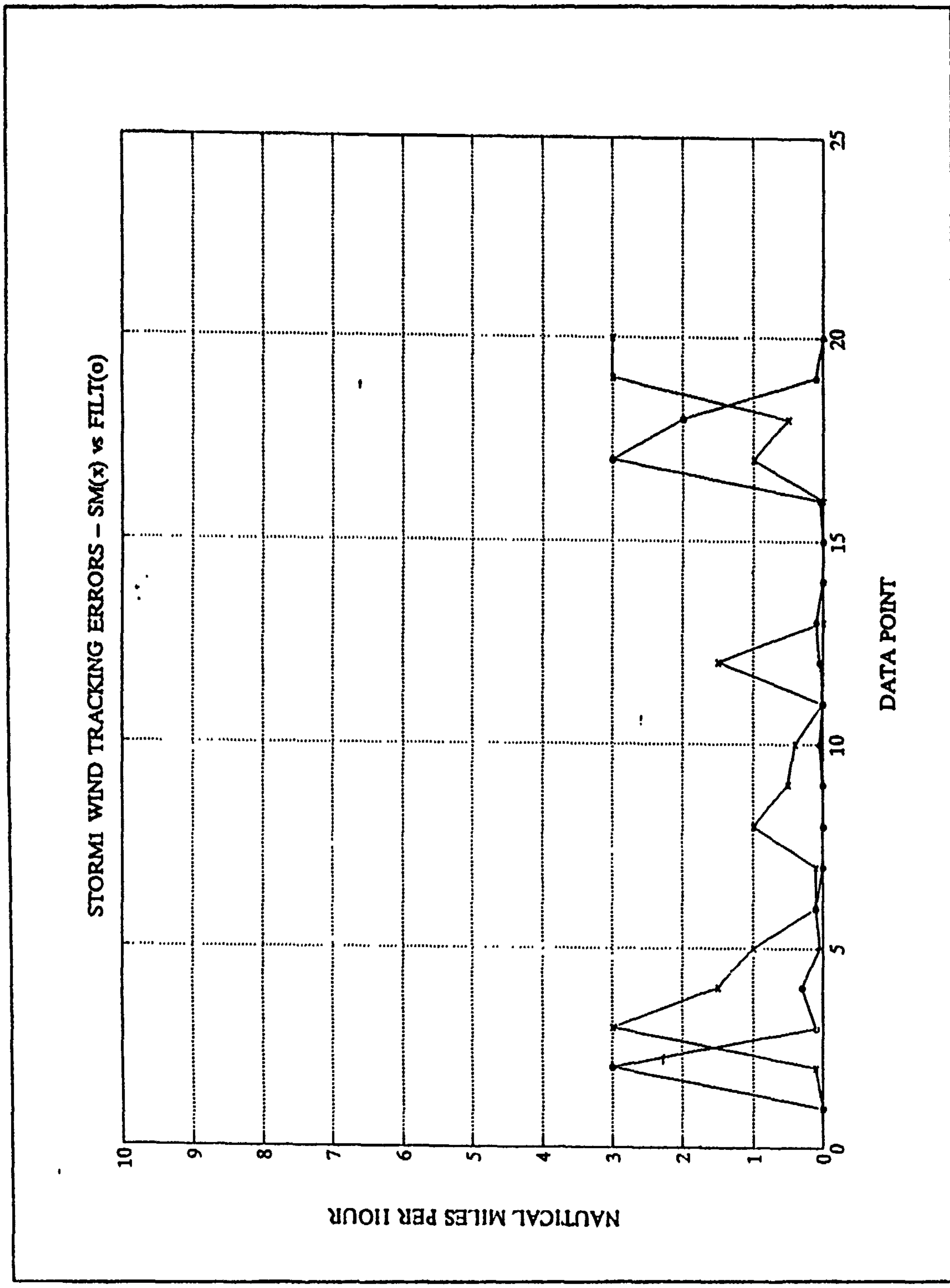

Figure 17. The Filter and Smoother Tracking Errors of Typhoon Pat

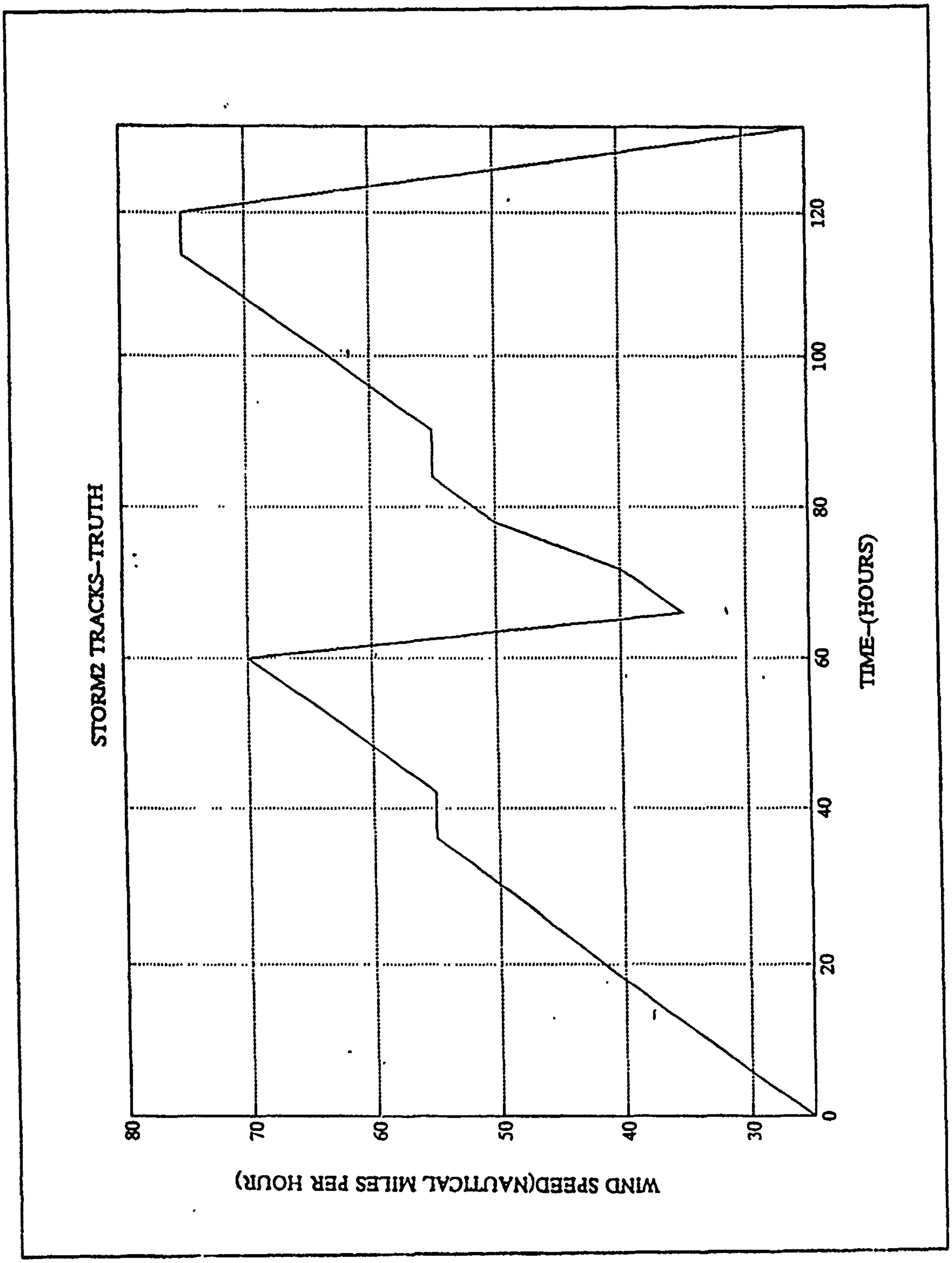

Figure 18. The Best Track Wind Speed of Typhoon Tess [Ref. 6]

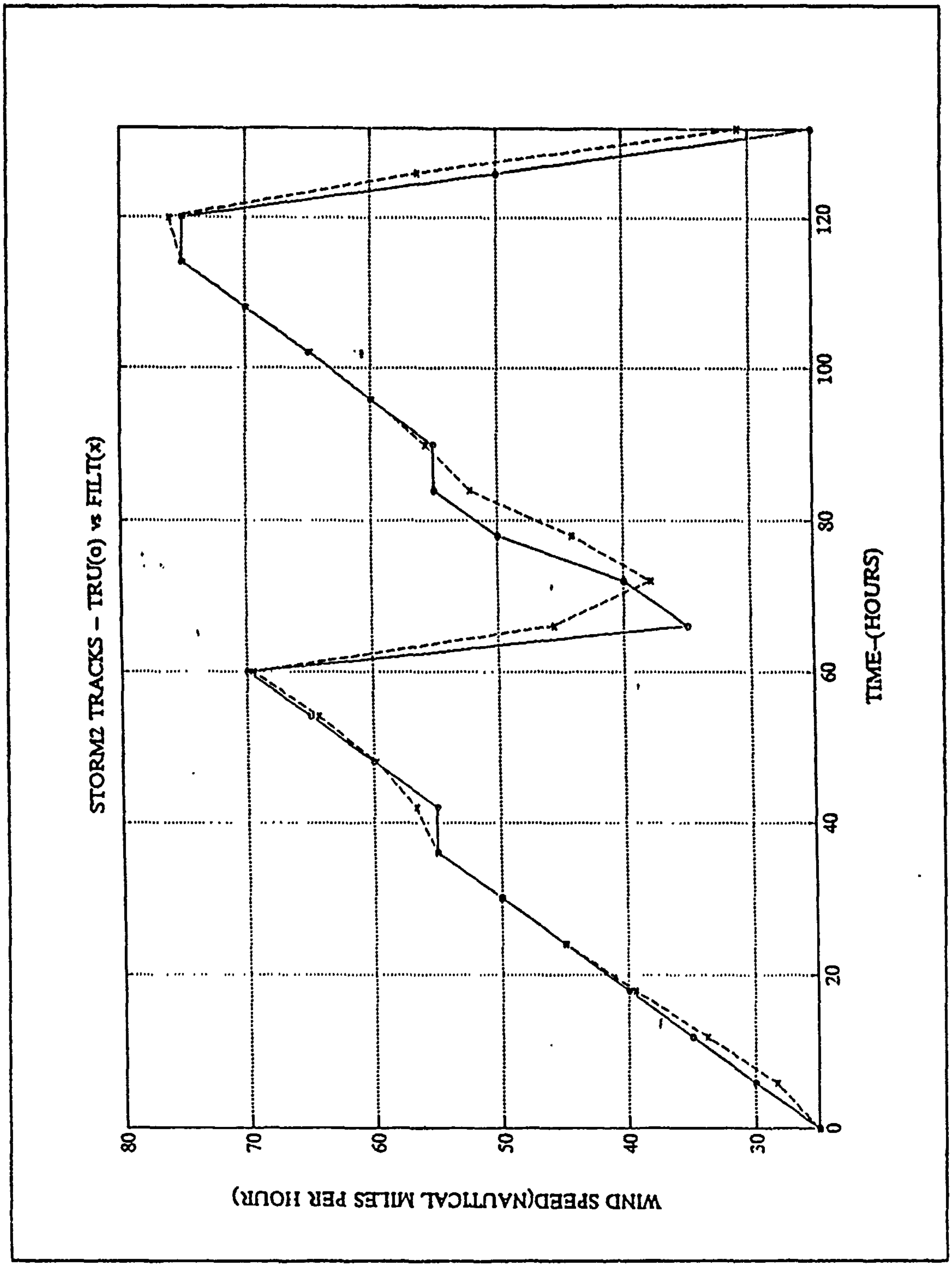

Figure 19. Filtered Track of Typhoon Tess' Best Track Wind Speed

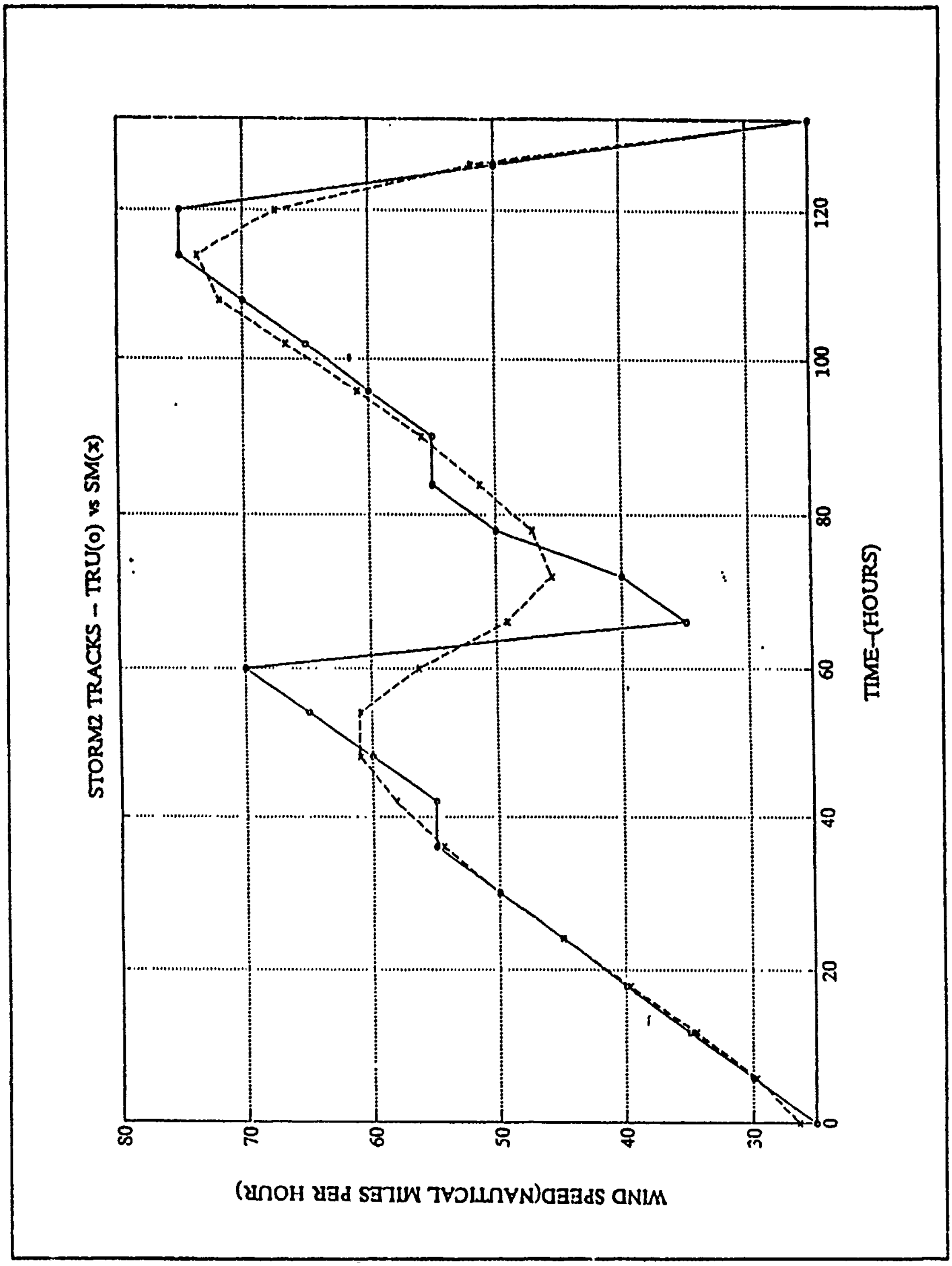

Figure 20. Smoothed Track of Typhoon Tess' Best Track Wind Speed

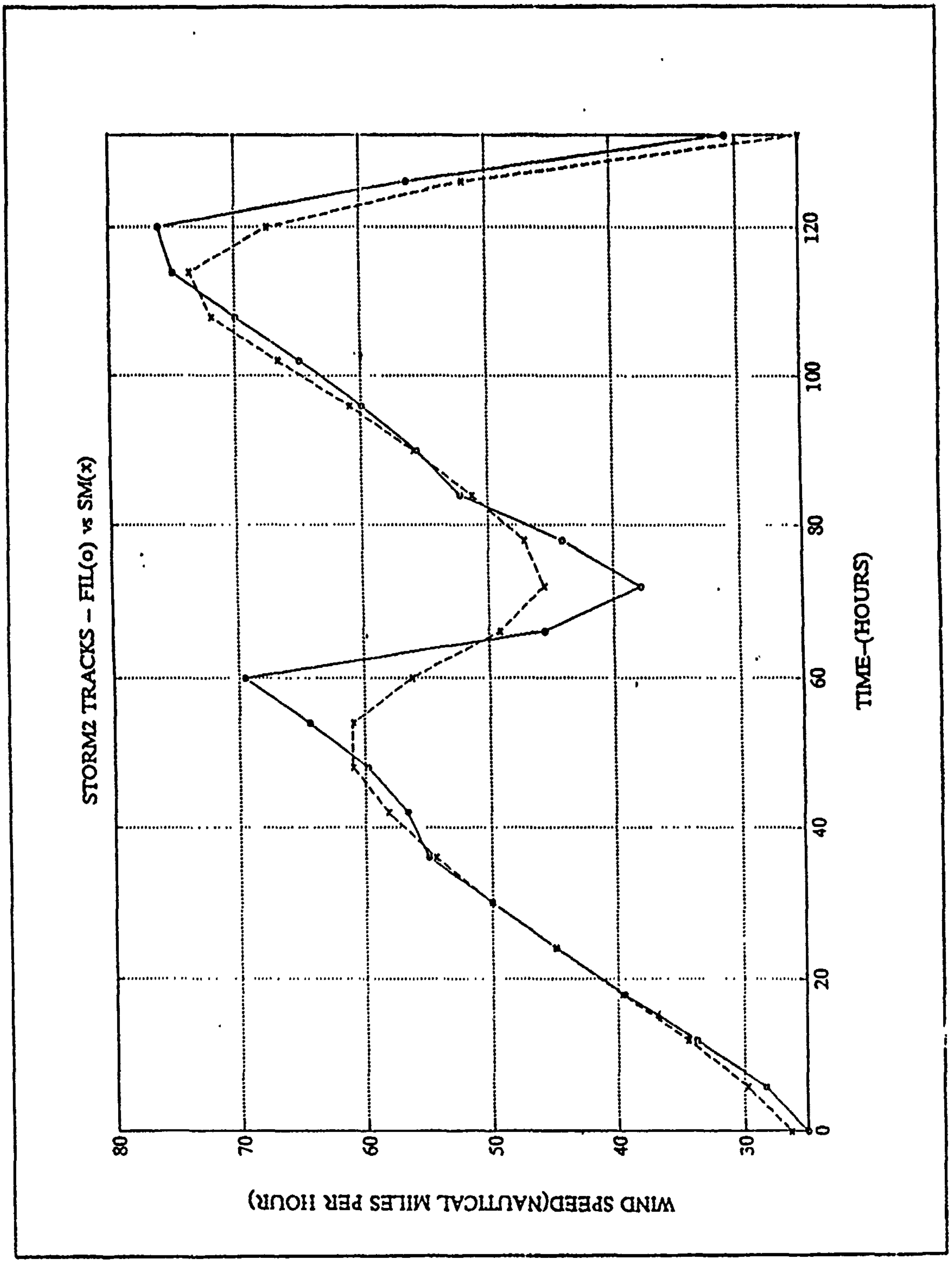

Figure 21. Filtered and Smoothed Track of Typhoon Tess' Best Track Wind Speed

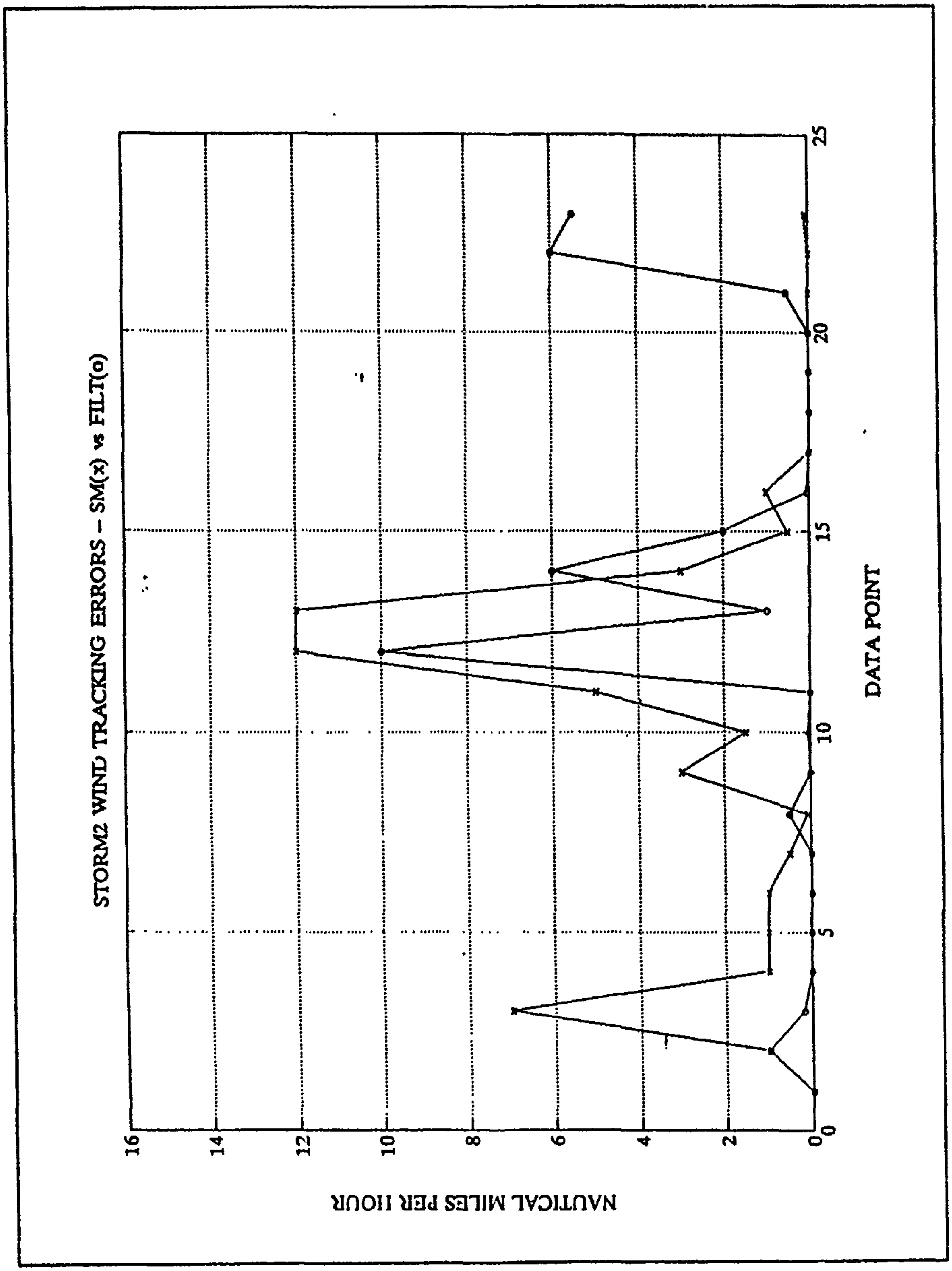

Figure 22. The Filter and Smoother Tracking Errors of Typhoon Tess

2. The Observed Wind Speed Data

There was uncertainity in the observed data obtained from the JTWC [Ref. 6]. This data has more than one data at the same time instant for the different positions from the eye of the hurricane. This is shown in Figures 23 and 24. There was a strong potential for the filter to go unstable. This was data smoothed using the Equations (4.8) and (4.9). The data obtained before and after curve fitting is shown in Figures 25 and 26.

$$H_k = \begin{bmatrix} 1.0 & -T_{-2} & T_{-2}^2 \\ 1.0 & -T_{-1} & T_{-1}^2 \\ 1.0 & 0.0 & 0.0 \\ 1.0 & T_1 & T_1^2 \\ 1.0 & T_2 & T_2^2 \end{bmatrix} \tag{4.8}$$

$$\hat{x}_k = [H_k^T H_k]^{-1} H_k^T z_k \tag{4.9}$$

where

z_k = measurements to be smoothed, and

$\hat{x}_k$ = smoothed measurements.

a. *Typhoon Pat*

Using the interpolated data as an observed data, tracking results obtained for typhoon Pat are shown in Figures 27 and 28. The filter and smoother estimates the wind speed accurately. There is no potential for the filter and smoother to go unstable. The accuracy of the filter is about 70%, and the smoother is about 65%. Due to the instant change in the wind speed, the smoother cannot adapted to this change easily.

b. *Typhoon Tess*

The performance of the filter and the smoother are better in Typhoon Tess. They estimate the wind speed very accurately. Again, there is no potential for the filter and smoother to go unstable. During the tracking scenario the filter gives the actual observed value and the smoother does improve the accuracy of these estimates. The tracking error is usually zero or very close to zero. The accuracy of the filter and smoother are almost the same in this hurricane which is about 85%. The tracking results are shown in Figures 29 and 30.

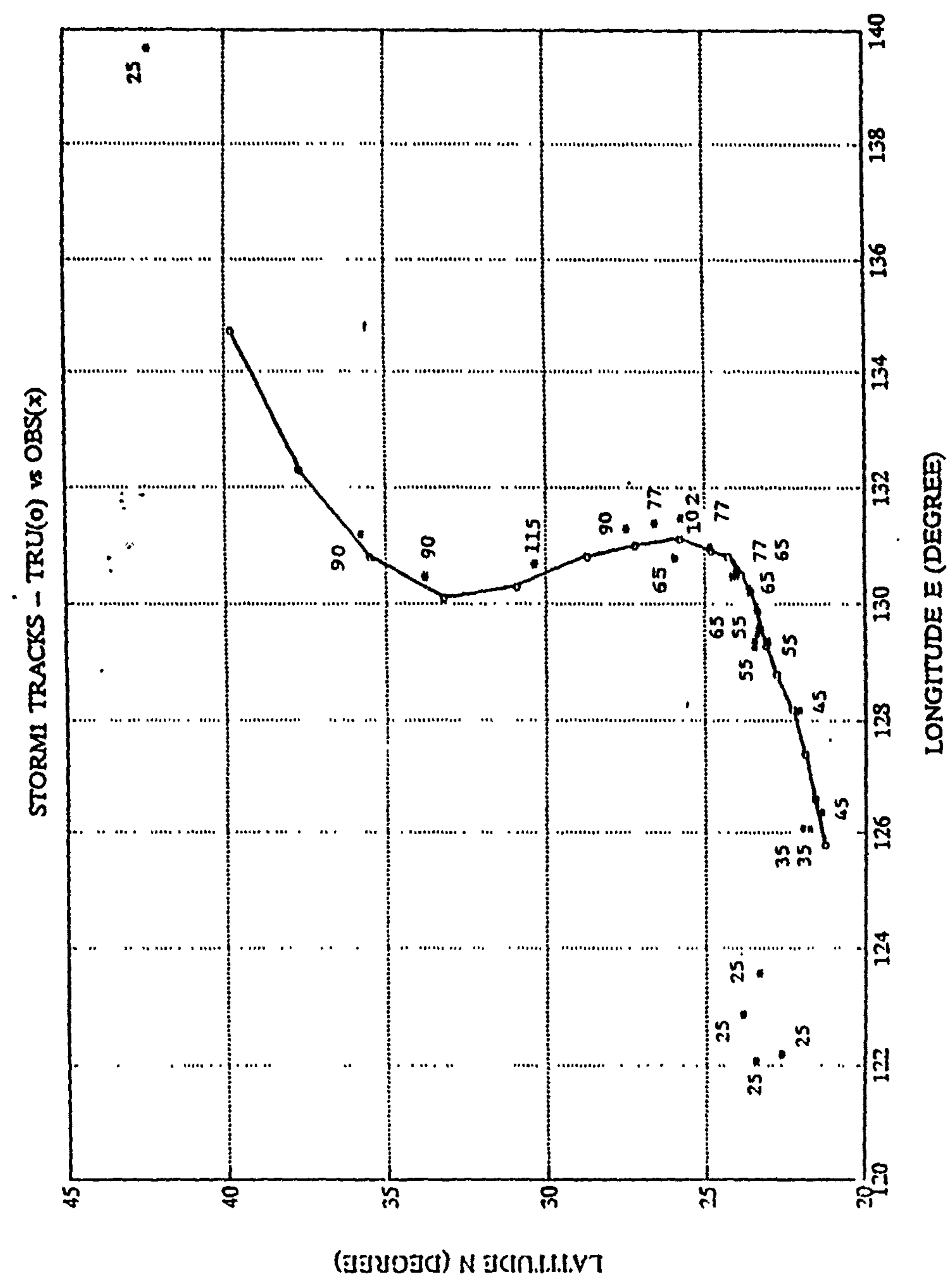

Figure 23. The Observed Wind Speed at Some Distance of Typhoon Pat [Ref. 6]

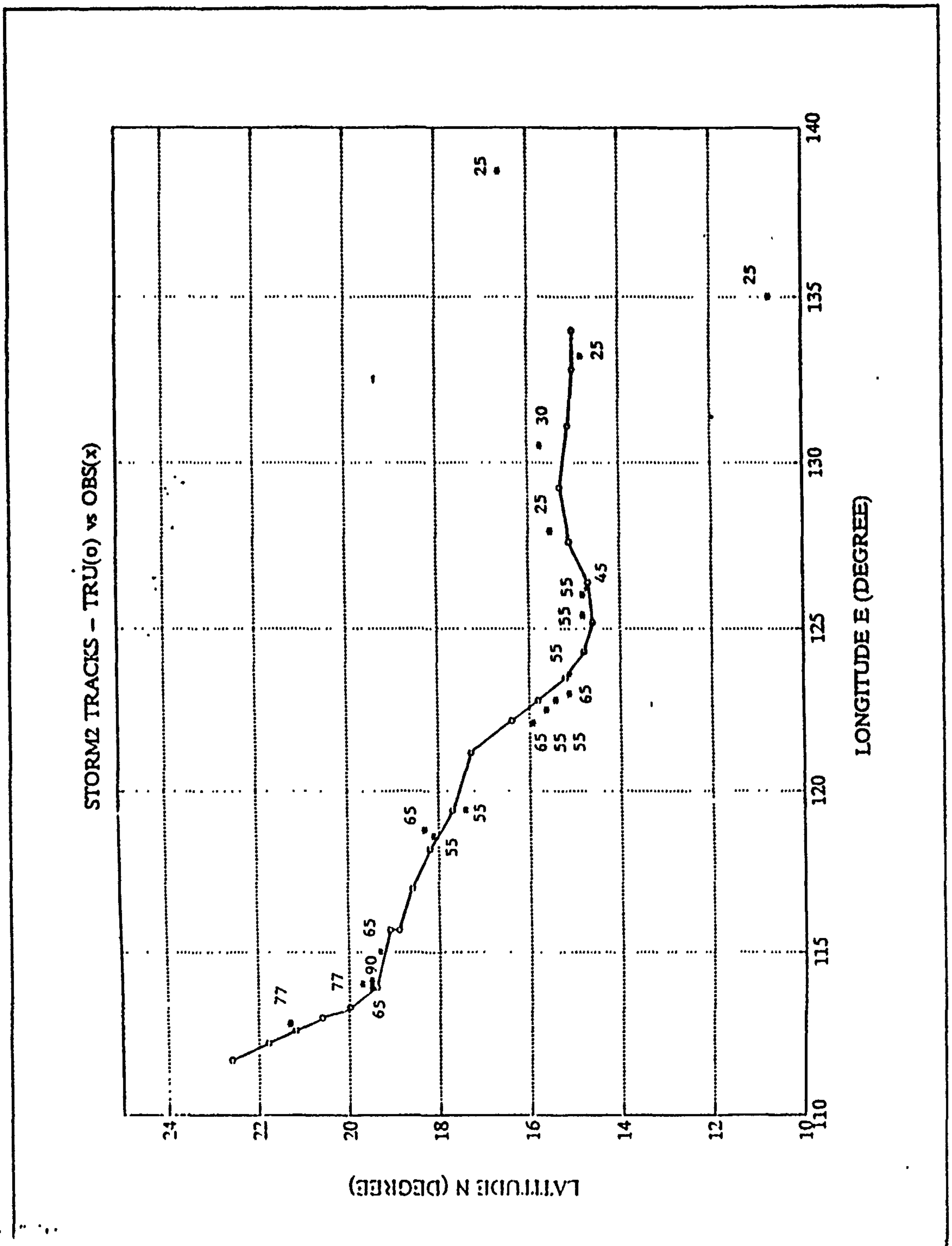

Figure 24. The Observed Wind Speed at Some Distance of Typhoon Tess [Ref. 6]

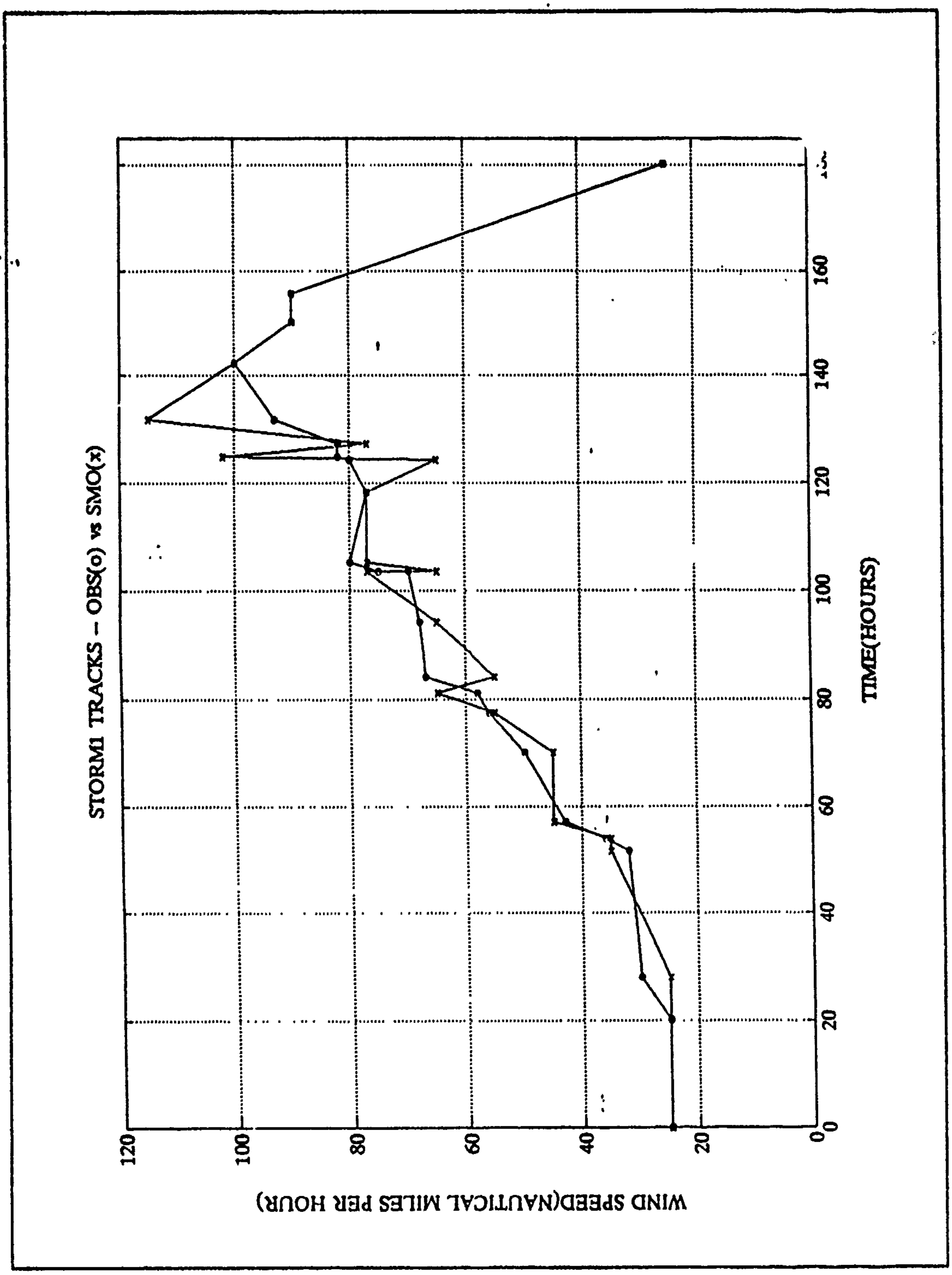

Figure 25. The Observed and Interpolated Track of Typhoon Pat

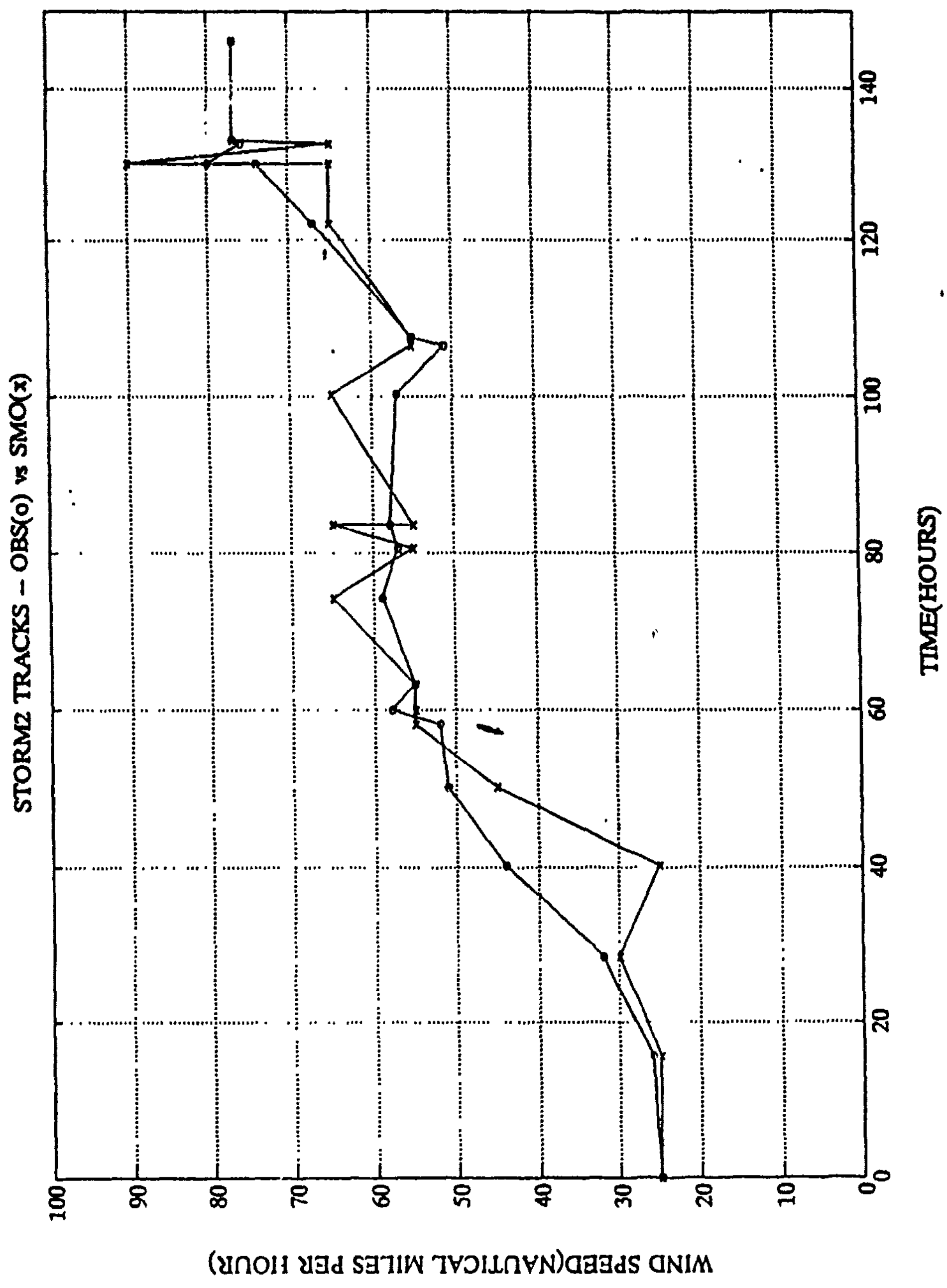

Figure 26. The Observed and Interpolated Track of Typhoon Tess

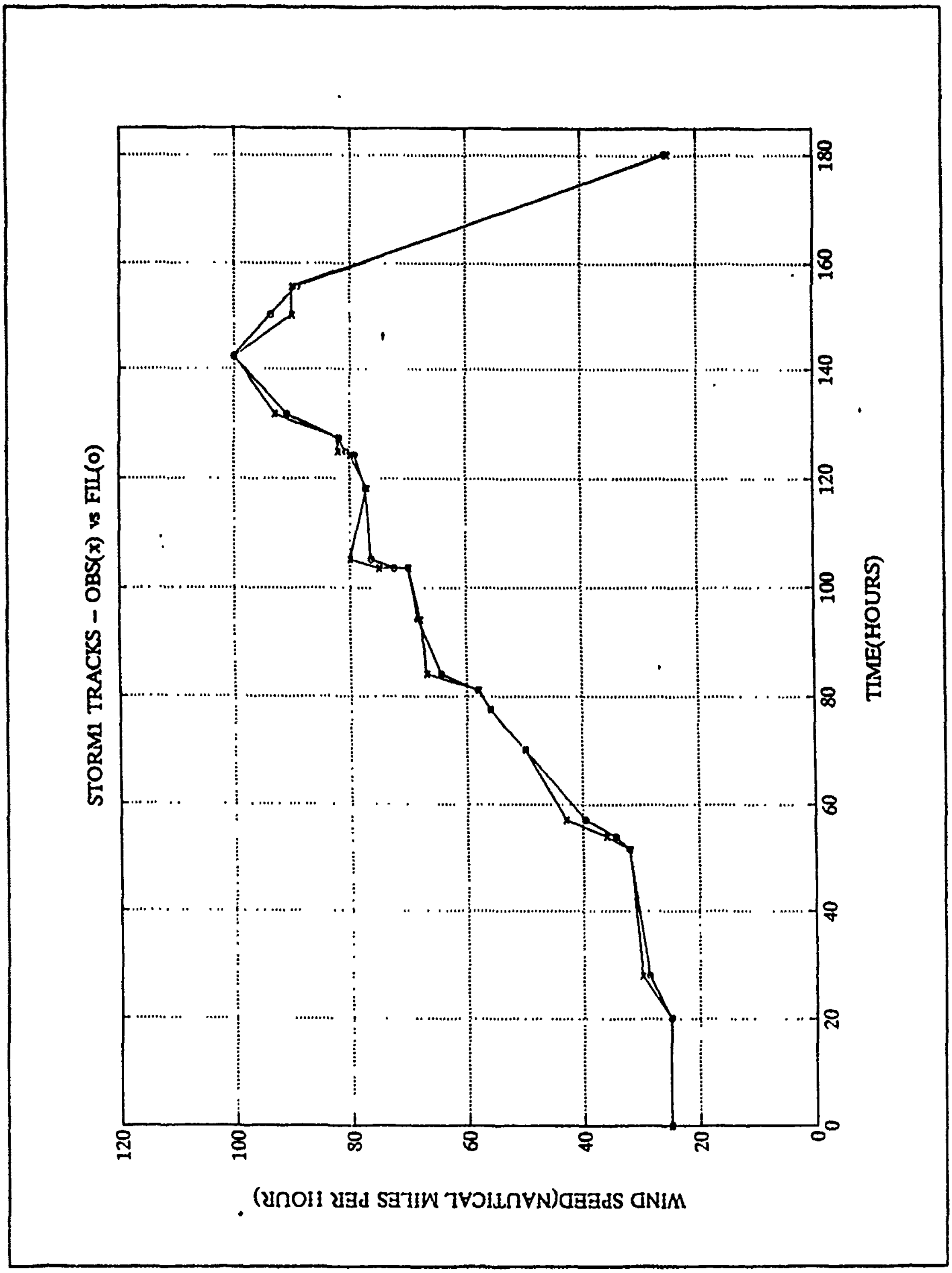

Figure 27. Filtered Track of Typhoon Pat's Observed Wind Speed

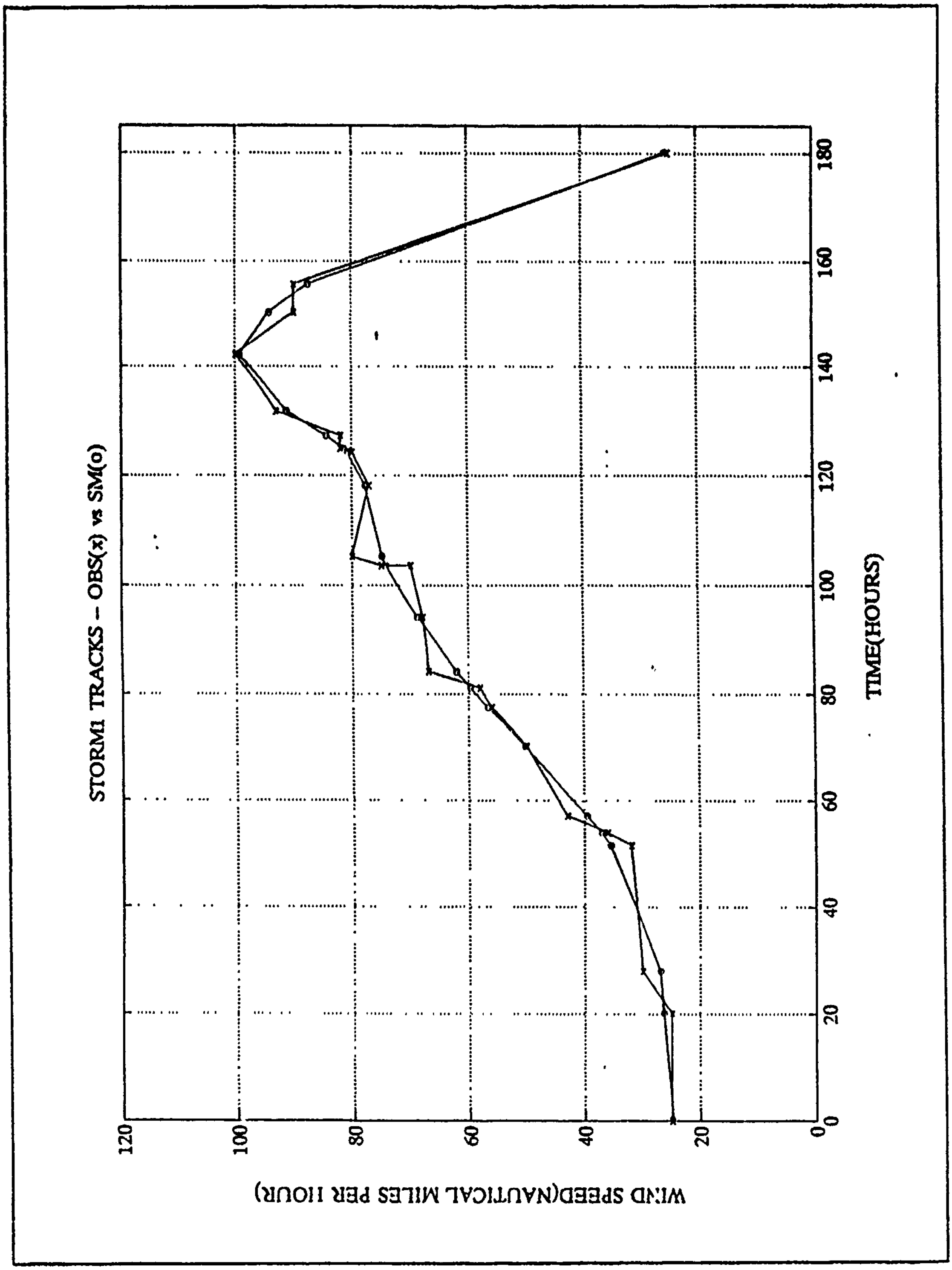

Figure 28. Smoothed Track of Typhoon Pat's Observed Wind Speed

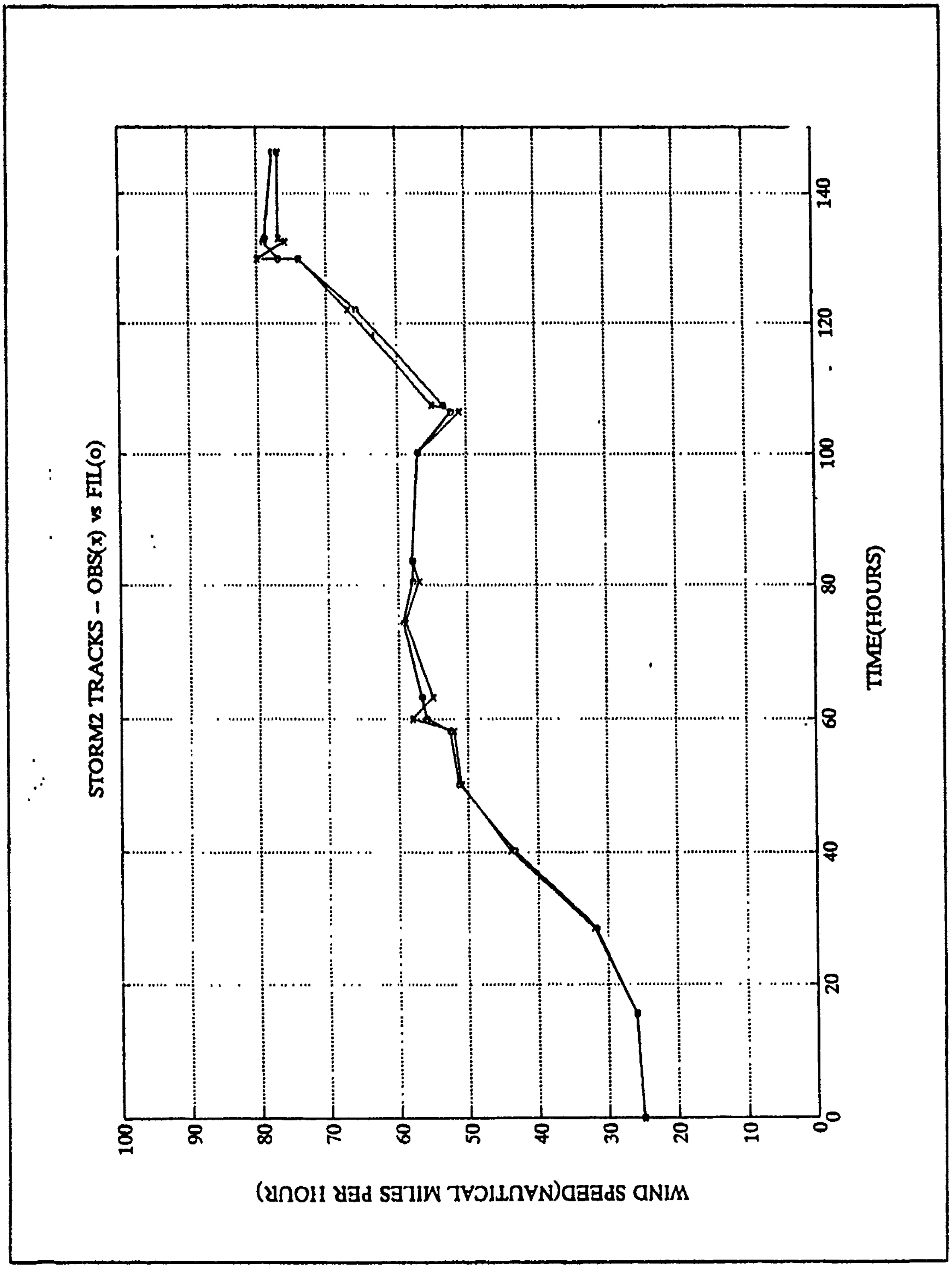

Figure 29. Filtered Track of Typhoon Tess' Observed Wind Speed

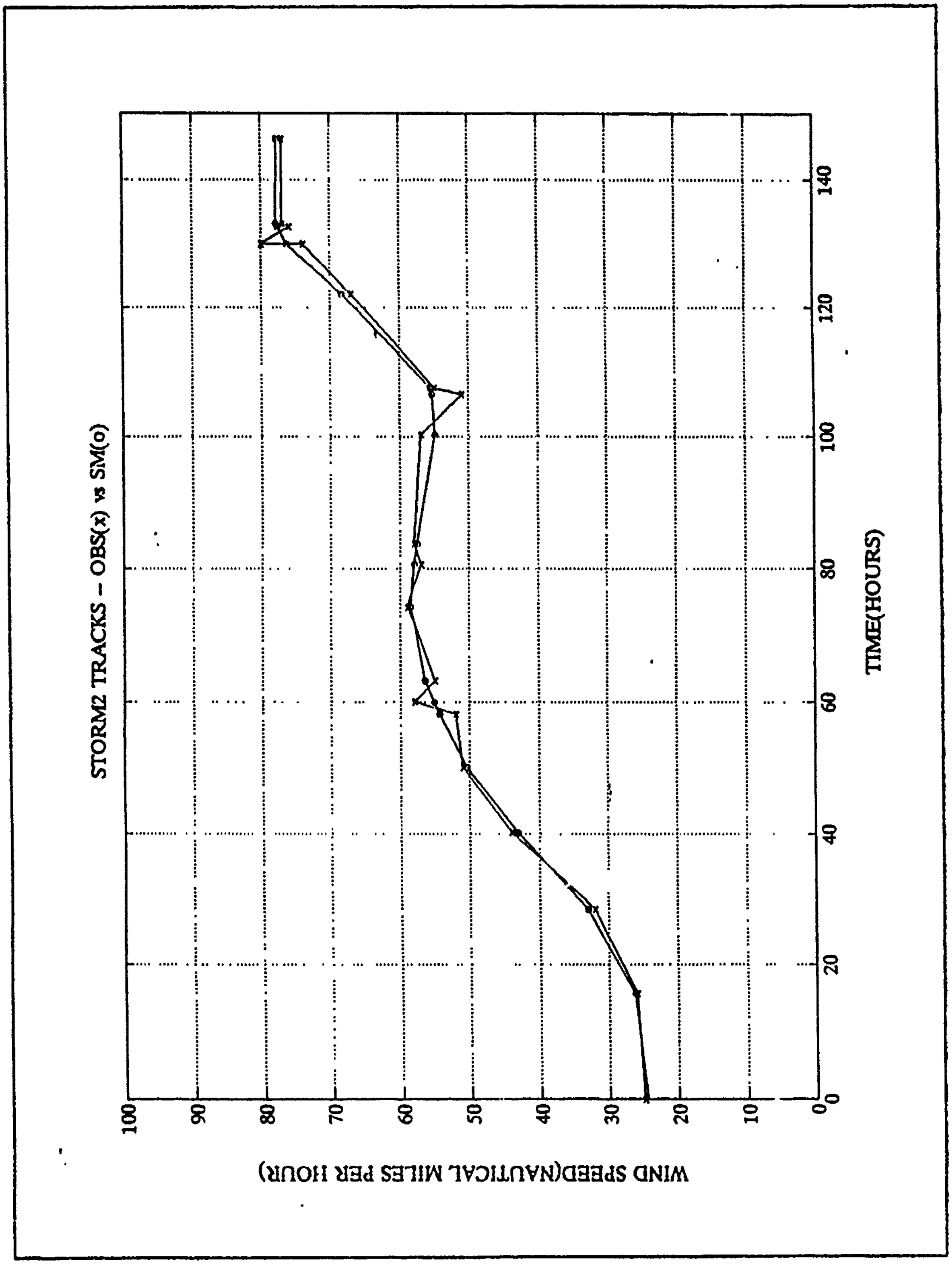

Figure 30. Smoothed Track of Typhoon Tess' Observed Wind Speed

V. CONCLUSIONS

The purpose of this research was to improve the accuracy and storm tracking capability of a Kalman filter tracking by implementing a fixed interval smoothing algorithm. Two different tropical storms were simulated and the accuracy of the observed, the filtered, and the smoothed storm tracks were analyzed and discussed.

The fixed interval smoothing algorithm improved the position accuracy of the storm in all of the tracking scenarios simulated. However, the smoothed result was not always the most accurate for every storm track. The smoother did improve the track accuracy on the basis of the best track storm positions. The effectiveness of the smoother increased as the storm lifetime increased and the storm course change decreased.

The storm wind speed tracking scheme implemented worked well. However, because this tracking involves the addition of a time-varying value of the state excitation matrix, Q_k, there was a strong potential for the filter to go unstable. This was observed during the storm wind speed tracking. It was difficult to decide the value of Q_k and R_k for observed wind speed tracking, because intensity could not be observed many times. This problem was solved by using a curve fitting method and then this data was used for inputs to the tracking problem. The results show that this method can be used to interpolate the uncertain data and to avoid an unstable filter.

The application of the Kalman filter tracker to the storm tracking problem would be very useful in attempting to predict the storm's track when little data is available, as seen in observed wind speed tracking problem. Then, by using the filter and smoothing algorithm, track of the storm's past history can be calculated allowing for a more accurate prediction of the storm's future track. There was no standard deviation for observed wind data. If JTWC can obtain standard deviations for observed wind data, this can be used. The wind data obtained has much missing data, some times causing an unstable filter.

APPENDIX A. STORM.FOR

This is a listing of the STORM.FOR program used to generate the data for the target tarcks presented in this thesis. In order to run this program, the STORM1.DAT or STORM2.DAT file must be available.

```
C  *** STORM1***

C*********** TO RUN *************
C
C      1) ENSURE STORM DATA IS AVAILABLE
C      2) RUN STORM1 OR STORM2
C      3) COPY OBSDATA,FILDATA, & SMDATA --> MATLAB SUB-DIR.
C      4) BEGIN MATLAB --> RUN STORM2.M
C
C  ************************************************************
C  THIS PROGRAM EMPLOYS AN ADAPTIVE EXTENDED KALMAN
C  FILTER WITH A FIXED INTERVAL SMOOTHING ALGORITHM TO TRACK A
C  TROPICAL STORM USING OBSERVED LATITUDES AND LONGITUDES.
C  ************************************************************

C  ***VARIABLE DEFINITIONS***

C       AK             =       SMOOTHING FILTER GAIN MATRIX
C       AKT            =       TRANSPOSE OF AK
C       BRG            =       MEASURED TARGET BEARING IN RADIANS
C       BRKKM1         =       PREDICTED TARGET BEARING MEASUREMENT
C                              IN RADIANS BRG(K|K-1)
C       DBRG           =       MEASURED TARGET BEARING IN DEGREES
C       DT             =       TIME DELAY BETWEEN OBSERVATIONS, T(K)
C                              - T(K1)
C       DTOR           =       DEGREE TO RADIAN CONVERSION FACTOR
C       E1,E2          =       MEASUREMENT RESIDUAL, Z(K) - H(X(K|K-1))
C       E1M1,E2M1      =       MEASUREMENT RESIDUAL AT PREVIOUS
C                              OBSERVATION
C       E1M2,E2M2      =       MEASUREMENT RESIDUAL TWO OBSERVATIONS
C                              PREVIOUS
C       FAC1           =       RECIPROCAL OF VARE
C       G              =       KALMAN GAIN VECTOR
C       GATE1          =       1.5*STANDARD DEVIATION OF RESIDUAL
C                              PROCESS, USED AS A GATE IN
C                              MANEUVER DETECTION
C       H              =       MEASUREMENT MATRIX
C       HDG            =       ESTIMATED TARGET HEADING IN DEGREES
C       HT             =       TRANSPOSE OF H
C       I              =       COUNTER
C       IMAT           =       4 X 4 IDENTITY MATRIX
```

```
C       J                =       COUNTER
C       K                =       ITERATION INTERVAL
C       LPKK             =       STATE COVARIANCE MATRIX AFTER PREVIOUS
C                                OBSERVATIONS
C       LPKKM1           =       A PRIORI STATE COVARIANCE ESTIMATE
C       LXKK             =       STATE ESTIMATE AFTER PREVIOUS OBSERVATIONS
C       LXKKM1           =       A PRIORI STATE ESTIMATE
C       M1,M2            =       AVERAGE OF RESIDUALS OVER LAST THREE
C                                OBSERVATIONS
C       PHI              =       DISCRETE-TIME STATE TRANSITION MATRIX
C       PHIT             =       TRANSPOSE OF PHI
C       PI               =       3.141592654
C       PKK              =       ESTIMATION ERROR COVARIANCE MATRIX, P(K|K)
C       PKKS             =       SMOOTHED ERROR COVARIANCE MATRIX
C       PKKM1            =       PREDICTED ESTIMATION ERROR COVARIANCE
C                                MATRIX, P(K|K-1)
C       PKKM1S           =       PREDICTED ERROR COVARIANCE MATRIX FOR
C                                SMOOTHING, P(K+1|K)
C       IPKKM1S          =       INVERSE OF PKKM1S
C       PSS              =       ERROR COVARIANCE MATRIX FOR
C                                SMOOTHING, P(K|K)
C       R                =       MEASUREMENT NOISE COVARIANCE
C       RANGE            =       DISTANCE FROM SENSOR TO A PRIORI TARGET
C                                POSITION
C       RTOD             =       RADIAN TO DEGREE CONVERSION FACTOR
C       SPD              =       ESTIMATED TARGET SPEED IN KNOTS
C       TEMP             =       TEMPORARY STORAGE MATRICES USED IN
C                                MATRIX    OPERATIONS
C       VARE             =       VARIANCE OF RESIDUALS PROCESS
C       XDIFF            =       DISTANCE IN X DIRECTION FROM SENSOR TO
C                                A PRIORI TARGET POSITION
C       XKK              =       ESTIMATED TARGET STATE VECTOR, X(K|K)
C       XKKS             =       SMOOTHED TARGET STATE VECTOR
C       XKKM1            =       PREDICTED TARGET STATE VECTOR,
C                                X(K|K-1)
C       XKKM1S           =       PREDICTED TARGET STATE VECTOR FOR
C                                SMOTHING, X(K+1|K)
C       XPOS             =       ESTIMATED TARGET POSITION IN X
C                                DIRECTION
C       XS               =       SENSOR POSITION IN X DIRECTION
C       XSS              =       TARGET STATE VECTOR FOR SMOOTHING,
C                                X(K|K)
C       XT               =       TRUE TARGET POSITION IN X DIRECTION
C       YDIFF            =       DISTANCE IN Y DIRECTION FROM SENSOR TO
C                                A PRIORI TARGET POSITION
C       YPOS             =       ESTIMATED TARGET POSITION IN Y DIRECTION
C       YS               =       SENSOR POSITION IN Y DIRECTION
C       YT               =       TRUE TARGET POSITION IN Y DIRECTION
C       ZX               =       OBSERVED POSITION IN X DIRECTION
C       ZY               =       OBSERVED POSITION IN Y DIRECTION

C VARIABLE DECLARATIONS
        CHARACTER*1 A,B

        REAL*4 XKK(4,1),XKKM1(4,1),LPKKM1(4,4),LXKKM1(4,1)
        REAL*4 H(2,4),HT(4,2),G(4,2),TEMP1(2,1),TEMP2(2,4),TEMP3(2,1)
```

```
      REAL*4 TEMP4(4,2),TEMP5(4,1),TEMP6(4,4),TEMP7(4,4)
      REAL*4 PKK(4,4),PKKM1(4,4),Z(2,1)
      REAL*4 LXKK(4,1),LPKK(4,4),XS(10),YS(10),DBRG(10),BRG
      REAL*4 PHI(4,4),PHIT(4,4),IMAT(4,4),XT,YT
      REAL*4 GATE1,E(2,1),VARE(2,2),IVARE(2,2)
      REAL*4 DT,DTF,XDIFF,YDIFF,RANGE,XS1,YS1,BRG1,BRKKM1
      REAL*4 DATE,HR,MN,LAT,LONG,TOTIM,TIME,TIMEM1,DATE1
      REAL*4 OBSERR(300),FAC1,SIGTH2,SIGVT2,R(2,2),ETOTAL,EAVG,RTOD
      REAL*4 X2,YS2,BRG2,ZX,ZY,M1,E1,E1M1,E1M2,DTOR,TRKERR(300)
      REAL*4 M2,E2,E2M1,E2M2,G11,G13,G21,G23,ZXM1,ZYM1
      REAL*4 XKKS(4,1,300),PKKS(4,4,300)
      REAL*4 XNNM1(4,1),XSS(4,1),XKKM1S(4,1)
      REAL*4 PNNM1(4,4),PSS(4,4),PKKM1S(4,4),IPKKM1S(4,4)
      REAL*4 AK(4,4),AKT(4,4),II(4,4),STRKERR(300),DTS(300)
      REAL*4 TEMP1S(4,4),TEMP2S(4,1),TEMP3S(4,1)
      REAL*4 TEMP4S(4,4),TEMP5S(4,4),TEMP6S(4,4)
      REAL*4 AS,ASA,ASL,NAV,MET
      INTEGER*2 NP
      INTEGER*, PCN
C OPEN OUTPUT DATA FILES
      OPEN(UNIT=2,FILE='STORM1.DAT',STATUS='OLD')
      OPEN(UNIT=3,FILE ='OUTDATA.DAT',STATUS='NEW')
      OPEN(UNIT=4,FILE='TRUDATA.DAT',STATUS='NEW')
      OPEN(UNIT=5,FILE='FILDATA.DAT',STATUS='NEW')
      OPEN(UNIT=6,FILE='SMDATA.DAT',STATUS='NEW')
      OPEN(UNIT=7,FILE='ELLIPDAT.DAT',STATUS='NEW')
      OPEN(UNIT=8,FILE='MATRIX.DAT',STATUS='NEW')
      OPEN(UNIT=9,FILE='ERRDATA.DAT',STATUS='NEW')
      OPEN(UNIT=10,FILE='ERRSDATA.DAT',STATUS='NEW')
C RADIAN/DEGREE CONVERSION FACTORS
      RTOD=57.29577951
      DTOR=0.01745293

C COMPUTE 4X4 IDENTITY MATRIX
      DO 5 I=1,4
      DO 5 J=1,4
      IF (I.EQ.J) THEN
              IMAT(I,J)=1.0
      ELSE
              IMAT(I,J)=0.0
      ENDIF
5     CONTINUE

      DO 6 I=1,2
       DO 6 J=1,4
          H(I,J)=0.0
6     CONTINUE
      H(1,1)=1.0
      H(2,3)=1.0

C INITIALIZE TIME COUNTER
      TOTTIM=0.0
      TIMEM1=0.0
      NP=0

C INITIALIZE COUNTER FOR MANEUVER GATE
```

```
      E1M1=0.0
      E1M2=0.0

C COMPUTE BEARING MEASUREMENT COVARIANCE
C     BEARING ERROR STANDARD DEVIATION = 1 NM
      WRITE(*,*) 'FILTERING OBSERVED DATA WITH KALMAN FILTER'
      WRITE(*,*) '***=====***'
810   READ(2,1001,END=800)DATE,HR,MN,LAT,A,LONG,B,PCN,NAV,MET
C SATELLITE DATA FOR MEASUREMENT NOISE COV. MATRIX VALUES
      IF(PCN.EQ.1)THEN
         AS=256.0
      ELSEIF(PCN.EQ.3)THEN
         AS=900.0
      ELSEIF(PCN.EQ.5)THEN
         AS=1600.0
C RADAR DATA
      ELSEIF(PCN.EQ.2)THEN
         AS=100.0
      ELSEIF(PCN.EQ.4)THEN
         AS=225.0
      ELSEIF(PCN.EQ.6)THEN
         AS=625.0
C AIRCRAFT DATA
      ELSE
         AS=((NAV)**2+(MET)**2)**0.5
      ENDIF
      R(1,1)=AS
      R(1,2)=0.0
      R(2,1)=0.0
      R(2,2)=AS


C ******************************************************************

C READ IN OBSERVATION PACKET (DATE,TIME,LAT,LONG)
C     DT=TIME(K)-TIME(K-1)


C     READ(2,1001,END=800)DATE,HR,MN,LAT,A,LONG,B
1001  FORMAT(F6.0,F2.0,F2.0,F3.0,A1,F4.0,A1,I1,2(F2.0))

      NP=NP+1

      MN=MN/60.0
      LAT=LAT/10
      LONG=LONG/10
      TIME=HR+MN
C     WRITE (3,1) DATE,HR,MN,LAT,A,LONG,B
1     FORMAT(1X,F7.0,4X,F3.0,1X,F6.4,6X,F4.1,A1,3X,F5.1,A1)

      IF (NP.EQ.1) THEN
         DATE1=DATE
         TIMEM1=TIME
      ENDIF
```

```
         IF (DATE.NE.DATE1) THEN
            TIME=TIME+24
            DT=TIME-TIMEM1
            TIME=TIME-24
         ELSE
            DT=TIME-TIMEM1
         ENDIF

         DTF=DT*60.0
         DTS(NP)=DT
         TOTTIM=TOTTIM+DT
C        WRITE (3,2) TIME,TOTTIM,DT
2        FORMAT(1X,F7.4,5X,F6.2,5X,F6.2)

         CALL FINDPHI(PHI,DT)

         Z(1,1)=LONG
         Z(2,1)=LAT
         ZX=LONG
         ZY=LAT

         IF(NP.EQ.1) THEN
                 CALL INIT(LONG,LAT,XKK,PKK)
C                WRITE(*,*)'X(0|0,0):'
                 DO 601 I=1,4
                  LXKK(I,1)=XKK(I,1)
C                 WRITE(3,*) '**************'
C                 WRITE(3,*) (XKK(I,1),I=1,4)
601              CONTINUE


C                WRITE(3,*)'P(0|0,0):'
                 DO 602 I=1,4
                  DO 602 J=1,4
C                 LPKK(I,J)=PKK(I,J)
C                 WRITE(3,401)PKK(I,J)
401               FORMAT(4F14.4)
602              CONTINUE

         ENDIF

C PROJECT AHEAD STATE AND ERROR COVARIANCE ESTIMATES
C        X(K+1|K) = PHI * X(K|K)
         CALL MATMUL(PHI,XKK,4,4,1,XKKM1)

C        WRITE(*,*)'X(',TIME,'|',TIMEM1,',0):'
         DO 603 I=1,4
C         WRITE(3,*) (XKKM1(I,1),I=1,4)
C         WRITE(3,*) '*****************'
          LXKKM1(I,1)=XKKM1(I,1)
603      CONTINUE



C        P(K+1|K) = (PHI * P(K|K) * PHIT) + Q
```

```
      CALL MATRAN(PHI,PHIT,4,4)
      CALL MATMUL(PHI,PKK,4,4,4,TEMP6)
      CALL MATMUL(TEMP6,PHIT,4,4,4,TEMP7)
      CALL GETQ(Q)
      CALL MATADD(TEMP7,Q,4,4,1,PKKM1)
      DO 408 I=1,4
      DO 408 J=1,4
              LPKKM1(I,J)=PKKM1(I,J)
408   CONTINUE

C     WRITE(*,*)'P(',TIME,'|',TIMEM1,',0):'
      DO 604 I=1,4
C      WRITE(3,402)(PKKM1(I,J),J=1,4)
402   FORMAT(4F14.4)
604   CONTINUE


204   CONTINUE


C COMPUTE OBSERVATION RESIDUAL
C    E=Z(K)-H*X(K|K-1)
        CALL MATMUL(H,XKKM1,2,4,1,TEMP1)
        CALL MATSUB(Z,TEMP1,2,1,E)

C COMPUTE VARIANCE OF RESIDUALS SEQUENCE
C AND ADAPTIVE GATE VALUE
C    VAR(E)=H*PKKM1*HT+R
        CALL MATRAN(H,HT,2,4)
        CALL MATMUL(H,PKKM1,2,4,4,TEMP2)
        CALL MATMUL(TEMP2,HT,2,4,2,TEMP3)
        CALL MATADD(TEMP3,R,2,2,1,VARE)
C       WRITE(3,*)'VARIANCE OF RESIDUALS = ',VARE
C       GATE1=1.5*SQRT(VARE)

C COMPUTE KALMAN GAIN MATRIX
C    G=PKKM1*HT*(H*PKKM1*HT+R)**-1
        CALL MATRAN(H,HT,2,4)
        CALL MATMUL(PKKM1,HT,4,4,2,TEMP4)
        CALL MATINV(VARE,2,IVARE)
        CALL MATMUL(TEMP4,IVARE,4,2,2,G)

C       WRITE(3,*)'PKKM1*HT ='
        DO 414 I=1,4
C        WRITE(3,*)TEMP4(I,1)
414     CONTINUE

C       WRITE(3,*)'G ='
        DO 613 I=1,4
C        WRITE(3,*)G(I,1)
613     CONTINUE

C       IF (L.EQ.1) THEN
C             G11=G(1,1)
C             G13=G(3,1)
C       ELSE
```

```
C               G21=G(1,1)
C               G23=G(3,1)
C         ENDIF

C COMPUTE UPDATED ESTIMATE
C    X(K|K)=X(K|K-1)+G*E, WHERE E=Z(K)-H*X(K|K-1)
          CALL MATMUL(G,E,4,2,1,TEMP5)
          CALL MATADD(TEMP5,XKKM1,4,1,1,XKK)

C         WRITE(3,*)'X(',TIME,'|',TIME,',',L,'):'
          DO 605 I=1,4
C          WRITE(3,*)XKK(I,1)
605       CONTINUE


C COMPUTE UPDATED ERROR COVARIANCE MATRIX
C    P(K|K)=(I - G*H)*P(K|K-1)
          CALL MATMUL(G,H,4,2,4,TEMP6)
          CALL MATSUB(IMAT,TEMP6,4,4,TEMP7)
          CALL MATMUL(TEMP7,PKKM1,4,4,4,PKK)

C         WRITE(3,*)'P(',TIME,'|',TIME,',',L,'):'
          DO 606 I=1,4
C          WRITE(3,406)(PKK(I,J),J=1,4)
406        FORMAT(4F14.4)
606       CONTINUE


C THESE STATEMENTS ARE FOR THE SMOOTHING ALGORITHM

                DO 620 I=1,4
                 XKKS(I,1,NP)=XKK(I,1)
620             CONTINUE

                DO 630 I=1,4
                 DO 630 J=1,4
                   PKKS(I,J,NP)=PKK(I,J)
630             CONTINUE

C COMPUTE TRUE TRACKING ERROR
        ASA=XKK(1,1)
        ASL=XKK(3,1)
        TRKERR(NP)=SQRT((LAT-ASA)**2+(LONG-ASL)**2)

C COMPUTE OBSERVATION ERROR
C       OBSERR(NP)=SQRT((ASLAT-ZX)**2+(ASLONG-ZY)**2)

C SAVE LATEST RESIDUALS FOR AVERAGING
C       E1=E

C COMPUTE THE AVERAGE RESIDUAL OVER THE PAST THREE OBSERVATIONS
C       M1=(E1+E1M1+E1M2)/3

C       WRITE(*,*)'PAST THREE RESIDUALS FOR SENSOR 1 ARE :',E1,E1M1,E1M2
C       WRITE(*,*)'BEARING AVERAGE OF SENSOR 1 = ',M1
C       WRITE(*,*)'MANEUVER GATE FOR SENSOR 1 = ',GATE1
```

```
C       E1M2=E1M1
C       E1M1=E1

C COMPUTE ERROR ELLIPSE DATA
C       CALL ELLIP(XKK(1,1),XKK(3,1),PKK(1,1),PKK(3,3),PKK(1,3))

C COMPUTE ESTIMATED X-Y POSITION, COURSE, AND SPEED
        XPOS=XKK(1,1)
        YPOS=XKK(3,1)
        IF (XKK(2,1).EQ.0 .AND. XKK(4,1).EQ.0) THEN
                HDG=0.0
        ELSE
                HDG=RTOD*ATAN2(XKK(2,1),XKK(4,1))
        ENDIF
        IF (HDG.LT.0.0) HDG=HDG+360
        SPD=60*SQRT(XKK(2,1)**2+XKK(4,1)**2)
C       WRITE(*,*) 'FILTERED DATA FOR DATA POINT',NP
        WRITE(3,*) 'FILTERED DATA FOR DATA POINT',NP
C       WRITE(*,*) 'TIME    X POS   Y POS  HEADING  SPEED'
        WRITE(3,*) 'TIME    X POS   Y POS  HEADING  SPEED'
C       WRITE(*,*)TOTTIM,XPOS,YPOS,HDG,SPD
        WRITE(3,*)TOTTIM,XPOS,YPOS,HDG,SPD
        WRITE(4,*)TOTTIM,ZX,ZY
        WRITE(5,*)TOTTIM,XPOS,YPOS,PKK(1,1)
        WRITE(9,*)NP,TRKERR(NP)
1002    FORMAT(1X,5F10.3)
1003    FORMAT(1X,F6.2,3X,F10.1,2X,F11.1,3X,F8.1,3X,F8.1)
1004    FORMAT(1X,F6.2,3(F8.1,2X))

C COMPARE BEARING ERRORS TO MANEUVER DETECTION GATES

        IF ((ABS(M1).GT.(GATE1))) THEN
             WRITE(*,*)'*** MANEUVER DETECTION ***'
C            WRITE(3,*)'*** MANEUVER DETECTION ***'
             CALL REINIT(DT,ZX,ZY,ZXM1,ZYM1,LPKKM1,XKKM1,PKKM1)
             E1M1=0.0
             E1M2=0.0
             GOTO 204
        ENDIF

        TIMEM1=TIME
        DATE1=DATE

        ZXM1=ZX
        ZYM1=ZY

        GOTO 810

C THIS IS WHERE THE SMOOTHING ALGORITHM STARTS
C FIXED INTERVAL SMOOTHING
800     WRITE(*,*) 'SMOOTHING FILTERED DATA WITH A'
        WRITE(*,*) 'FIXED INTERVAL SMOOTHING ALGORITHM'
        WRITE(*,*) '***=====***'
```

```
        DO 1000 KK=1,NP-1
         K=NP-KK

         DT=DTS(K+1)

         TIME=TIMEM1-DT
         CALL FINDPHI(PHI,DT)

         DO 901 I=1,4
          XSS(I,1)=XKKS(I,1,K)
901      CONTINUE

         DO 902 I=1,4
          DO 902 J=1,4
           PSS(I,J)=PKKS(I,J,K)
902      CONTINUE

C CALCULATE THE PREDICTED STATE AND ERROR COVARIANCE MATRICES
C    X(K+1|K)=PHI*X(K|K)
         CALL MATMUL (PHI,XSS,4,4,1,XKKM1S)
C    P(K+1|K)=PHI*P(K|K)*PHIT+Q
         CALL MATRAN (PHI,PHIT,4,4)
         CALL MATMUL(PHI,PSS,4,4,4,TEMP6)
         CALL MATMUL(TEMP6,PHIT,4,4,4,TEMP7)
         CALL GETQ(Q)
         CALL MATADD(TEMP7,Q,4,4,1,PKKM1S)


C CALCULATE THE SMOOTHING FILTER GAIN MATRIX
C    AK=P(K|K)*PHIT*INV°P(K+1|K)
         CALL MATINV (PKKM1S,4,IPKKM1S)
         CALL MATMUL (PKKM1S,IPKKM1S,4,4,4,II)
         CALL MATMUL (PSS,PHIT,4,4,4,TEMP1S)
         CALL MATMUL (TEMP1S,IPKKM1S,4,4,4,AK)

         DO 904 I=1,4
           XNNM1(I,1)=XKKS(I,1,K+1)
904      CONTINUE

C CALCULATE THE SMOOTHED STATE ESTIMATE
C    XKKS=X(K|K)+AK*(X(K+1|N)-X(K+1|K)
         CALL MATSUB (XNNM1,XKKM1S,4,1,TEMP2S)
         CALL MATMUL (AK,TEMP2S,4,4,1,TEMP3S)
         CALL MATADD (XSS,TEMP3S,4,1,K,XKKS)

         DO 906 I=1,4
          DO 906 J=1,4
            PNNM1(I,J)=PKKS(I,J,K+1)
906      CONTINUE

C CALCULATE THE SMOOTHED COVARIANCE MATRIX
C    PKKS=P(K|K)+AK*[P(K+1|N)-P(K+1|K)]*AKT
         CALL MATSUB (PNNM1,PKKM1S,4,4,TEMP4S)
         CALL MATRAN (AK,AKT,4,4)
         CALL MATMUL (AK,TEMP4S,4,4,4,TEMP5S)
```

```
          CALL MATMUL (TEMP5S,AKT,4,4,4,TEMP6S)
          CALL MATADD (PSS,TEMP6S,4,4,K,PKKS)

C COMPUTE ESTIMATED X-Y POSITION, COURSE, AND SPEED
        SXPOS=XKKS(1,1,K)
        SYPOS=XKKS(3,1,K)
        IF (XKKS(2,1,K).EQ.0 .AND. XKKS(4,1,K).EQ.0) THEN
                SHDG=0.0
        ELSE
                SHDG=RTOD*ATAN2(XKKS(2,1,K),XKKS(4,1,K))
        ENDIF
        IF (SHDG.LT.0.0) SHDG=SHDG+360
        SSPD=60*SQRT(XKKS(2,1,K)**2+XKKS(4,1,K)**2)
C       WRITE(*,*) 'SMOOTHED DATA FOR DATA POINT',K
        WRITE(3,*) 'SMOOTHED DATA FOR DATA POINT',K
C       WRITE(*,*) 'TIME    X POS   Y POS  HEADING  SPEED'
        WRITE(3,*) 'TIME    X POS   Y POS  HEADING  SPEED'
C       WRITE(*,*)TOTTIM,SXPOS,SYPOS,SHDG,SSPD
        WRITE(3,*)TOTTIM,SXPOS,SYPOS,SHDG,SSPD
1010    FORMAT(1X,5F10.3)
1020    FORMAT(1X,F6.2,3X,F10.1,2X,F11.1,3X,F8.1,3X,F8.1)
1030    FORMAT(1X,F6.2,3(F8.1,2X))

        TIMEM1=TIME
1000    CONTINUE

C       CLOSE(UNIT=4)

C CALCULATE THE SMOOTHED TRACKING ERROR
C       OPEN(UNIT=4,FILE='TRUDATA.DAT',STATUS='OLD')
        DO 1100 K=1,NP
         SXPOS=XKKS(1,1,K)
         SYPOS=XKKS(3,1,K)
C        READ(4,1001)DATE,HR,MN,LAT,A,LONG,B,PCN
         STRKERR(K)=SQRT((LAT-SXPOS)**2+(LONG-SYPOS)**2)
         WRITE(6,1120)K,SXPOS,SYPOS,PKKS(1,1,K)

         WRITE(10,*)K,STRKERR(K)
1100    CONTINUE
1110    FORMAT(I4,2F8.1)
1120    FORMAT(I4,3(F8.1,2X))
1130    FORMAT(I4,3F8.1)

        CLOSE(UNIT=2)
        CLOSE(UNIT=3)
        CLOSE(UNIT=4)
        CLOSE(UNIT=5)
        CLOSE(UNIT=6)
        CLOSE(UNIT=7)
        CLOSE(UNIT=8)
        CLOSE(UNIT=9)
        CLOSE(UNIT=10)

        WRITE(*,*) 'FILTERED & SMOOTHED OUTPUT DATA IS LOCATED IN THE'
        WRITE(*,*) 'DATA FILE OUTDATA.DAT.  FOR GRAPHIC RESULTS,'
        WRITE(*,*) 'ENSURE OBSDATA.DAT, FILDATA.DAT, & SMDATA.DAT ARE'
```

```
      WRITE(*,*) 'IN THE MATLAB SUB-DIRECTORY AND RUN THE MATLAB'
      WRITE(*,*) 'M-FILE STORM2.M'
      STOP
      END

C*****************************************************************
C                              SUBROUTINES
C*****************************************************************


      SUBROUTINE FINDPHI(PHI,DT)
C ********************************************************
C     COMPUTES THE VALUES OF THE PHI MATRIX
C ********************************************************
      REAL*4 PHI(4,4),DT

      DO 1501 I=1,4
      DO 1501 J=1,4
      DO 1501 K=1,2
            PHI(I,J)=0.0
1501  CONTINUE

C COMPUTE PHI MATRIX
      DO 1500 I=1,4
       PHI(I,I)=1.0
1500  CONTINUE
      PHI(1,2)=DT
      PHI(3,4)=DT

      RETURN

      END


      SUBROUTINE INIT(LONG,LAT,XKK,PKK)
C ***************************************************
C     THIS ROUTINE INITIALIZES THE STATE
C     AND ERROR COVARIANCE ESTIMATES
C ***************************************************
      REAL*4 XKK(4,1),PKK(4,4)
      REAL*4 LAT,LONG

C INITIAL STATE ESTIMATE
      XKK(3,1)=LAT
      XKK(2,1)=0.0
      XKK(1,1)=LONG
      XKK(4,1)=0.0

C INITIAL ERROR COVARIANCE ESTIMATE
      PKK(1,1)=100.0
      PKK(1,2)=0.0
      PKK(1,3)=0.0
      PKK(1,4)=0.0
      PKK(2,1)=0.0
      PKK(2,2)=0.025
```

```
        PKK(2,3)=0.0
        PKK(2,4)=0.0
        PKK(3,1)=0.0
        PKK(3,2)=0.0
        PKK(3,3)=100
        PKK(3,4)=0.0
        PKK(4,1)=0.0
        PKK(4,2)=0.0
        PKK(4,3)=0.0
        PKK(4,4)=0.025

                RETURN

                END

        SUBROUTINE GETQ(Q)
C**********************************************************
C       ROUTINE TO GET Q MATRIX
C**********************************************************
        REAL*4 Q(4,4)

        DO 100 I=1,4
         DO 100 J=1,4
100     Q(I,J)=0.0
        DO 200 I=1,4
200     Q(I,I)=100.

        RETURN

        END

        SUBROUTINE REINIT(DT,ZX,ZY,ZXM1,ZYM1,LPKKM1,XKKM1,PKKM1)
C *************************************************************
C       THIS ROUTINE RE-INITIALIZES THE STATE AND ERROR
C       COVARIANCE ESTIMATES
C *************************************************************
        REAL*4 DT,XKKM1(4,1),PKKM1(4,4)
        REAL*4 ZX,ZY,ZXM1,ZYM1,LPKKM1(4,4)

        XDIFF=ZX-ZXM1
        YDIFF=ZY-ZYM1

        XKKM1(1,1)=ZX
        XKKM1(2,1)=XDIFF/DT
        XKKM1(3,1)=ZY
        XKKM1(4,1)=YDIFF/DT

C       WRITE(3,*)'REINITIALIZED STATES ARE:'
        DO 100 I=1,4
C               WRITE(3,*)XKKM1(I,1)
100     CONTINUE

        PKKM1(1,1)=2.25*LPKKM1(1,1)
        PKKM1(1,2)=0.0
        PKKM1(1,3)=2.25*LPKKM1(1,3)
```

```
      PKKM1(1,4)=0.0
      PKKM1(2,1)=0.0
      PKKM1(2,2)=0.1111
      PKKM1(2,3)=0.0
      PKKM1(2,4)=0.0
      PKKM1(3,1)=2.25*LPKKM1(3,1)
      PKKM1(3,2)=0.0
      PKKM1(3,3)=2.25*LPKKM1(3,3)
      PKKM1(3,4)=0.0
      PKKM1(4,1)=0.0
      PKKM1(4,2)=0.0
      PKKM1(4,3)=0.0
      PKKM1(4,4)=0.1111

               RETURN

               END

      SUBROUTINE MP(XS1,YS1,XS2,YS2,BRG1,BRG2,ZX,ZY)
C ******************************************************************
C     THIS ROUTINE COMPUTES THE ESTIMATED
C      X,Y POSITION OBTAINED FROM MEASUREMENTS
C ******************************************************************
      REAL*4 ZX,ZY
      REAL*4 XS1,YS1,XS2,YS2,BRG1,BRG2
      REAL*4 NUMER,DENOM

C INITIAL STATE ESTIMATE

      NUMER=(-YS2*TAN(BRG2))+(YS1*TAN(BRG1))+XS2-XS1
      DENOM=TAN(BRG1)-TAN(BRG2)

      ZY=NUMER/DENOM
      ZX=(ZY-YS1)*TAN(BRG1)+XS1

      RETURN

      END


      SUBROUTINE ELLIP(XT,YT,P1,P3,P13)
C ***********************************************************
C     THIS SUBROUTINE COMPUTES ERROR ELLIPSE DATA
C     FROM ERROR COVARIANCE DATA
C ***********************************************************
C     DIMENSIONS AND DECLARATIONS
      REAL*4 XT,YT,XP(21),YP(21),A,B,THE1,SIG2X,SIG2Y
      REAL*4 SX,SY,PT,CT,ST,P1,P13,P3

      A=2*P13
      B=P1-P3
      THE1=0.5*ATAN2(A,B)
      A=(P1+P3)/2
      B=0.0
      IF (P13.EQ.0.0) GOTO 10
```

```
      B=P13/SIN(2.0*THE1)
10    SIG2X=ABS(A+B)
      SIG2Y=ABS(A-B)
      SX=SIG2X**0.5
      SY=SIG2Y**0.5
      PT=3.141592654/10
      CT=COS(THE1)
      ST=SIN(THE1)

      DO 100 IE=1,21
            XP(IE)=SX*COS(PT*IE)*CT-SY*SIN(PT*IE)*ST+XT
            YP(IE)=SX*COS(PT*IE)*ST+SY*SIN(PT*IE)*CT+YT
            WRITE(7,*)XP(IE),CHAR(9),YP(IE)
100   CONTINUE

      RETURN

      END


      SUBROUTINE MATMUL(A,B,L,M,N,C)
C *****************************************************************
C     THIS ROUTINE MULTIPLIES TWO MATRICES TOGETHER
C      ° C(L,N) = A(L,M) * B(M,N)
C *****************************************************************
C     DIMENSIONS AND DECLARATIONS
      REAL*4 A(L,M),B(M,N),C(L,N)

      DO 10 I=1,L
      DO 10 J=1,N
       C(I,J)=0.0
10    CONTINUE

      DO 100 I= 1,L
      DO 100 J= 1,N
      DO 100 K= 1,M
       C(I,J) = C(I,J) + A(I,K)*B(K,J)
100   CONTINUE

      RETURN

      END


      SUBROUTINE MATRAN(A,B,N,M)
C ****************************************************
C     THIS ROUTINE TRANSPOSES A MATRIX
C            ° B(M,N) = A'(N,M)
C ****************************************************
C     DIMENSIONS AND DECLARATIONS
      REAL*4 A(N,M), B(M,N)

      DO 100 I= 1,N
      DO 100 J= 1,M
       B(J,I) = A(I,J)
100   CONTINUE
```

```
      RETURN

      END


      SUBROUTINE MATSCL(Q,A,N,M,C)
C ****************************************************************
C     THIS ROUTINE MULTIPLIES A MATRIX WITH A SCALAR
C      ° C(N,M) = Q * A(N,M)
C ****************************************************************
C     DIMENSIONS AND DECLARATIONS
            REAL*4 A(N,M), C(N,M), Q

      DO 100 I = 1,N
      DO 100 J = 1,M
       C(I,J) = Q*A(I,J)
100   CONTINUE

      RETURN

      END


      SUBROUTINE MATSUB(A,B,N,M,C)
C *********************************************************
C     THIS ROUTINE SUBTRACTS TWO MATRICES
C    ° C(N,M) = A(N,M) - B(N,M)
C *********************************************************
C     DIMENSIONS AND DECLARATIONS
      REAL*4  A(N,M),B(N,M),C(N,M)

      DO 100 I = 1,N
      DO 100 J = 1,M
       C(I,J)=A(I,J)-B(I,J)
100   CONTINUE

      RETURN

      END

      SUBROUTINE MATADD(A,B,N,M,L,C)
C*********************************************************
C     THIS ROUTINE ADDS TWO MATRICES
C      ° C(N,M) = A(N,M) + B(N,M)
C*********************************************************
C     DIMENSIONS AND DECLARATIONS
      REAL*4  A(N,M),B(N,M),C(N,M,L)
      DO 100 I = 1,N
      DO 100 J = 1,M
       C(I,J,L)=A(I,J)+B(I,J)
100   CONTINUE

      RETURN
      END
```

```
        SUBROUTINE MATINV (A,N,C)
C*************************************************
C       THIS ROUTINE COMPUTES THE INVERSE OF
C       A MATRIX
C          C(N,N)=INV [A(N,N)]
C*************************************************
C         DIMENSIONS AND DECLARATIONS
          REAL*4 A(N,N),C(N,N),D(20,20)
          DO 100 I = 1,N
           DO 100 J = 1,N
100             D(I,J)=A(I,J)

          DO 115 I=1,N
           DO 115 J=N+1,2*N
115       D(I,J)=0.0

          DO 120 I=1,N
           J=I+N
120       D(I,J)=1.0

          DO 240 K=1,N
           M=K+1
           IF (K.EQ.N) GOTO 180
           L=K
           DO 140 I=M,N
140        IF (ABS(D(I,K)).GT.ABS(D(L,K))) L=I
           IF (L.EQ.K) GOTO 180

           DO 160 J=K,2*N
            TEMP=D(K,J)
            D(K,J)=D(L,J)
160        D(L,J)=TEMP

180        DO 185 J=M,2*N
185        D(K,J)=D(K,J)/D(K,K)

           IF (K.EQ.1) GOTO 220
           M1=K-1
           DO 200 I=1,M1
            DO 200 J=M,2*N
200       D(I,J)=D(I,J)-D(I,K)*D(K,J)

          IF (K.EQ.N) GOTO 260

220       DO 240 I=M,N
           DO 240 J=M,2*N
240          D(I,J)=D(I,J)-D(I,K)*D(K,J)

260       DO 265 I=1,N
           DO 265 J=1,N
            K=J+N
265       C(I,J)=D(I,K)

          RETURN
          END
```

APPENDIX B. WIND.FOR

This a listing of the WIND.FOR micro computer program used to generate the data for the storm wind speed tracks presented in this thesis. In order to run this program, the WINDO1.DAT file must be available.

```
C  ****************************************************************************

C  ***VARIABLE DEFINITIONS***

C       AK            =      SMOOTHING FILTER GAIN MATRIX
C       AKT           =      TRANSPOSE OF AK
C       BRG           =      MEASURED TARGET BEARING IN RADIANS
C       BRKKM1        =      PREDICTED TARGET BEARING MEASUREMENT IN RADIANS
C                                    BRG(K|K-1)
C       DBRG          =      MEASURED TARGET BEARING IN DEGREES
C       DT            =      TIME DELAY BETWEEN OBSERVATIONS,T(K) - T(K1)
C       DTOR          =      DEGREE TO RADIAN CONVERSION FACTOR
C       E1,E2         =      MEASUREMENT RESIDUAL, Z(K) - H(X(K|K-1))
C       E1M1,E2M1     =      MEASUREMENT RESIDUAL AT PREVIOUS OBSERVATION
C       E1M2,E2M2     =      MEASUREMENT RESIDUAL TWO OBSERVATIONS PREVIOUS
C       FAC1          =      RECIPROCAL OF VARE
C       G             =      KALMAN GAIN VECTOR
C       GATE1         =      1.5*STANDARD DEVIATION OF RESIDUAL PROCESS, USED AS A
C                                    GATE IN MANEUVER DETECTION
C       H             =      MEASUREMENT MATRIX
C       HDG           =      ESTIMATED TARGET HEADING IN DEGREES
C       HT            =      TRANSPOSE OF H
C       I             =      COUNTER
C       IMAT          =      4 X 4 IDENTITY MATRIX
C       J             =      COUNTER
C       K             =      ITERATION INTERVAL
C       LPKK          =      STATE COVARIANCE MATRIX AFTER PREVIOUS OBSERVATIONS
C       LPKKM1        =      A PRIORI STATE COVARIANCE ESTIMATE
C       LXKK          =      STATE ESTIMATE AFTER PREVIOUS OBSERVATIONS
C       LXKKM1        =      A PRIORI STATE ESTIMATE
C       M1,M2         =      AVERAGE OF RESIDUALS OVER LAST THREE OBSERVATIONS
C       PHI           =      DISCRETE-TIME STATE TRANSITION MATRIX
C       PHIT          =      TRANSPOSE OF PHI
C       PI            =      3.141592654
C       PKK           =      ESTIMATION ERROR COVARIANCE MATRIX, P(K|K)
C       PKKS          =      SMOOTHED ERROR COVARIANCE MATRIX
C       PKKM1         =      PREDICTED ESTIMATION ERROR COVARIANCE MATRIX, P(K|K-1
C       PKKM1S        =      PREDICTED ERROR COVARIANCE MATRIX FOR SMOOTHING, P(K+
C       IPKKM1S       =      INVERSE OF PKKM1S
C       PSS           =      ERROR COVARIANCE MATRIX FOR SMOOTHING, P(K|K)
C       R             =      MEASUREMENT NOISE COVARIANCE
C       RANGE         =      DISTANCE FROM SENSOR TO A PRIORI TARGET POSITION
C       RTOD          =      RADIAN TO DEGREE CONVERSION FACTOR
C       SPD           =      ESTIMATED TARGET SPEED IN KNOTS
C       TEMP          =      TEMPORARY STORAGE MATRICES USED IN MATRIX
```

```
C                                        OPERATIONS
C       VARE           =       VARIANCE OF RESIDUALS PROCESS
C       XDIFF          =       DISTANCE IN X DIRECTION FROM SENSOR TO A PRIORI
C                              TARGET POSITION
C       XKK            =       ESTIMATED TARGET STATE VECTOR, X(K|K)
C       XKKS           =       SMOOTHED TARGET STATE VECTOR
C       XKKM1          =       PREDICTED TARGET STATE VECTOR, X(K|K-1)
C       XKKM1S         =       PREDICTED TARGET STATE VECTOR FOR SMMOTHING, X(K+1
C       XPOS           =       ESTIMATED TARGET POSITION IN X DIRECTION
C       XS             =       SENSOR POSITION IN X DIRECTION
C       XSS            =       TARGET STATE VECTOR FOR SMOOTHING, X(K|K)
C       XT             =       TRUE TARGET POSITION IN X DIRECTION
C       YDIFF          =       DISTANCE IN Y DIRECTION FROM SENSOR TO A PRIORI
C                              TARGET POSITION
C       YPOS           =       ESTIMATED TARGET POSITION IN Y DIRECTION
C       YS             =       SENSOR POSITION IN Y DIRECTION
C       YT             =       TRUE TARGET POSITION IN Y DIRECTION
C       ZX             =       OBSERVED POSITION IN X DIRECTION
C       ZY             =       OBSERVED POSITION IN Y DIRECTION

C VARIABLE DECLARATIONS
        CHARACTER*1 A,B

        REAL*4 XKK(2,1),XKKM1(2,1),LPKKM1(2,2),LXKKM1(2,1)
        REAL*4 H(2,2),HT(2,2),G(2,1),TEMP1(2,1),TEMP2(2,2),TEMP3(2,1)
        REAL*4 TEMP4(2,2),TEMP5(2,1),TEMP6(2,2),TEMP7(2,2)
        REAL*4 PKK(2,2),PKKM1(2,2),Z(1,1)
        REAL*4 LXKK(2,1),LPKK(2,2),XS(10),YS(10),DBRG(10),BRG
        REAL*4 PHI(2,2),PHIT(2,2),IMAT(2,2),XT,YT
        REAL*4 GATE1,E(2,1),VARE(2,2),IVARE(2,2)
        REAL*4 DT,DTF,XDIFF,YDIFF,RANGE,XS1,YS1,BRG1,BRKKM1
        REAL*4 DATE,HR,MN,LAT,LONG,TOTIM,TIME,TIMEM1,DATE1
        REAL*4 OBSERR(300),FAC1,SIGTH2,SIGVT2,R(2,2),ETOTAL,EAVG,RTOD
        REAL*4 X2,YS2,BRG2,ZX,ZY,M1,E1,E1M1,E1M2,DTOR,TRKERR(300)
        REAL*4 M2,E2,E2M1,E2M2,G11,G13,G21,G23,ZXM1,ZYM1
        REAL*4 XKKS(2,1,300),PKKS(2,2,300)
        REAL*4 XNNM1(2,1),XSS(2,1),XKKM1S(2,1)
        REAL*4 PNNM1(2,2),PSS(2,2),PKKM1S(2,2),IPKKM1S(2,2)
        REAL*4 AK(2,2),AKT(2,2),II(2,2),STRKERR(300),DTS(300)
        REAL*4 TEMP1S(2,2),TEMP2S(2,1),TEMP3S(2,1)
        REAL*4 TEMP4S(2,2),TEMP5S(2,2),TEMP6S(2,2)
        REAL*4 AS,ASA,ASL,WIND,WINDD,NAV,MET
        INTEGER*2 NP,ASIM,K
        INTEGER*, PCN
C OPEN OUTPUT DATA FILES
        OPEN(UNIT=2,FILE='WINDO1.DAT',STATUS='OLD')
        OPEN(UNIT=3,FILE ='OUTDATA.DAT',STATUS='NEW')
        OPEN(UNIT=4,FILE='OBSDATA.DAT',STATUS='NEW')
        OPEN(UNIT=5,FILE='FILDATA.DAT',STATUS='NEW')
        OPEN(UNIT=6,FILE='SMDATA.DAT',STATUS='NEW')
        OPEN(UNIT=8,FILE ='MATRIX.DAT',STATUS='NEW')
        OPEN(UNIT=9,FILE ='PALDATA.DAT',STATUS='NEW')
C RADIAN/DEGREE CONVERSION FACTORS
        RTOD=57.29577951
        DTOR=0.01745293
```

```
C COMPUTE 4X4 IDENTITY MATRIX
        DO 5 I=1,2
        DO 5 J=1,2
        IF (I.EQ.J) THEN
                IMAT(I,J)=1.0
        ELSE
                IMAT(I,J)=0.0
        ENDIF
5       CONTINUE

        H(1,1)=1.0
        H(1,2)=0.0


C INITIALIZE TIME COUNTER
        TOTTIM=0.0
        TIMEM1=0.0
        WIND=0.0
        NP=0

C INITIALIZE COUNTER FOR MANEUVER GATE
        E1M1=0.0
        E1M2=0.0

C COMPUTE BEARING MEASUREMENT COVARIANCE
C       BEARING ERROR STANDARD DEVIATION = 1 NM
        WRITE(*,*) 'FILTERING OBSERVED DATA WITH KALMAN FILTER'
        WRITE(*,*) '***=====***'
810     READ(2,1001,END=800)DATE,HR,MN,LAT,A,LONG,B,PCN,WINDD,NAV,MET
C RADAR DATA FOR MEASUREMENT NOISE COV. MATRIX
        IF(PCN.EQ.1)THEN
           AS=100.0
        ELSEIF(PCN.EQ.2)THEN
           AS=225.0
        ELSEIF(PCN.EQ.3)THEN
           AS=625.0
        ELSEIF(PCN.EQ.4)THEN
           AS=900.0
C AIRCRAFT DATA
        ELSE
           AS=((NAV)**2+(MET)**2)**0.5
        ENDIF
        R(1,1)=AS
        R(1,2)=0.0
        R(2,1)=0.0
        R(2,2)=AS


C ****************************************************************

C READ IN OBSERVATION PACKET (DATE,TIME,LAT,LONG)
C       DT=TIME(K)-TIME(K-1)


1001    FORMAT(F6.0,F2.0,F2.0,F3.0,A1,F4.0,A1,I1,F3.0,2(F2.0))
```

```
        MN=MN/60.0
        LAT=LAT/10
        LONG=LONG/10
        TIME=HR+MN
1       FORMAT(1X,F7.0,4X,F3.0,1X,F6.4,6X,F4.1,A1,3X,F5.1,A1)
        NP=NP+1
        IF (NP.EQ.1) THEN
           DATE1=DATE
           TIMEM1=TIME
        ENDIF
        IF (DATE.NE.DATE1) THEN
           TIME=TIME+24
           DT=TIME-TIMEM1
           TIME=TIME-24
        ELSE
           DT=TIME-TIMEM1
        ENDIF

        DTF=DT*60.0
        DTS(NP)=DT
        TOTTIM=TOTTIM+DT
C       WRITE (*,*) DT,NP,WINDD
2       FORMAT(1X,F7.4,5X,F6.2,5X,F6.2)

        CALL FINDPHI(PHI,DT)

        Z(1,1)=WINDD
        ZY=WINDD

        IF(NP.EQ.1) THEN
        CALL INIT(WINDD,XKK,PKK)
C               WRITE(*,*)'X(0|0,0):'
                DO 601 I=1,2
                 LXKK(I,1)=XKK(I,1)
C                WRITE(3,*) '**************'
C                WRITE(*,*) (XKK(I,1),I=1,2)
601             CONTINUE
                 WRITE(*,*) '**************'
                 WRITE(*,*) ZY


C               WRITE(3,*)'P(0|0,0):'
                DO 602 I=1,2
                 DO 602 J=1,2
C                LPKK(I,J)=PKK(I,J)
C                WRITE(3,401)PKK(I,J)
401              FORMAT(2F14.4)
602             CONTINUE

        ENDIF

C PROJECT AHEAD STATE AND ERROR COVARIANCE ESTIMATES
C       X(K+1|K) = PHI * X(K|K)
        CALL MATMUL(PHI,XKK,2,2,1,XKKM1)
```

```
        DO 603 I=1,2
         LXKKM1(I,1)=XKKM1(I,1)
603     CONTINUE


C       P(K+1|K) = (PHI * P(K|K) * PHIT) + Q

        CALL MATRAN(PHI,PHIT,2,2)
        CALL MATMUL(PHI,PKK,2,2,2,TEMP6)
        CALL MATMUL(TEMP6,PHIT,2,2,2,TEMP7)
        CALL GETQ(Q,DT)
        CALL MATADD(TEMP7,Q,2,2,1,PKKM1)
        DO 408 I=1,2
        DO 408 J=1,2
               LPKKM1(I,J)=PKKM1(I,J)
408     CONTINUE


204     CONTINUE


C COMPUTE OBSERVATION RESIDUAL
C    E=Z(K)-H*X(K|K-1)
          IF(WINDD.EQ.0)THEN
          E(1,1)=0.0
          E(2,1)=0.0
          ELSE
          CALL MATMUL(H,XKKM1,2,2,1,TEMP1)
          CALL MATSUB(Z,TEMP1,2,1,E)
          ENDIF
C COMPUTE VARIANCE OF RESIDUALS SEQUENCE
C AND ADAPTIVE GATE VALUE
C    VAR(E)=H*PKKM1*HT+R
          CALL MATRAN(H,HT,2,2)
          CALL MATMUL(H,PKKM1,2,2,2,TEMP2)
          CALL MATMUL(TEMP2,HT,2,2,2,TEMP3)
          CALL MATADD(TEMP3,R,2,2,1,VARE)
C         WRITE(3,*)'VARIANCE OF RESIDUALS = ',VARE
C         GATE1=1.5*SQRT(VARE)

C COMPUTE KALMAN GAIN MATRIX
C    G=PKKM1*HT*(H*PKKM1*HT+R)**-1
          CALL MATRAN(H,HT,2,2)
          CALL MATMUL(PKKM1,HT,2,2,2,TEMP4)
          CALL MATINV(VARE,2,IVARE)
          CALL MATMUL(TEMP4,IVARE,2,2,1,G)


C COMPUTE UPDATED ESTIMATE
C    X(K|K)=X(K|K-1)+G*E, WHERE E=Z(K)-H*X(K|K-1)
          CALL MATMUL(G,E,2,2,1,TEMP5)
          CALL MATADD(TEMP5,XKKM1,2,1,1,XKK)
```

```
C         WRITE(3,*)'X(',TIME,'|',TIME,',',L,'):'
          DO 605 I=1,2
C          WRITE(3,*)XKK(I,1)
605       CONTINUE


C COMPUTE UPDATED ERROR COVARIANCE MATRIX
C    P(K|K)=(I - G*H)*P(K|K-1)
          CALL MATMUL(G,H,2,2,2,TEMP6)
          CALL MATSUB(IMAT,TEMP6,2,2,TEMP7)
          CALL MATMUL(TEMP7,PKKM1,2,2,2,PKK)


C THESE STATEMENTS ARE FOR THE SMOOTHING ALGORITHM

                DO 620 I=1,2
                 XKKS(I,1,NP)=XKK(I,1)
C          WRITE(*,*)XKKS(I,1,NP),PKKS(I,1,NP)
620             CONTINUE

                DO 630 I=1,2
                 DO 630 J=1,2
                   PKKS(I,J,NP)=PKK(I,J)
630             CONTINUE

        WRITE(3,*) 'FILTERED DATA FOR DATA POINT',NP
        WRITE(3,*) 'TIME    VEL.   ACCELL.   HEADING  SPEED'
        WRITE(3,*)TOTTIM,XKK(1,1),XKK(2,1)
        WRITE(4,*)TOTTIM,ZY
        WRITE(5,*)TOTTIM,XKK(1,1),XKK(2,1),PKK(1,1)
        WRITE(9,*)NP
1002    FORMAT(1X,5F10.3)
1003    FORMAT(1X,F6.2,3X,F10.1,2X,F11.1,3X,F8.1,3X,F8.1)
1004    FORMAT(1X,F6.2,3(F8.1,2X))

C COMPARE BEARING ERRORS TO MANEUVER DETECTION GATES

        IF ((ABS(M1).GT.(GATE1))) THEN
C            WRITE(*,*)'*** MANEUVER DETECTION ***'
C            WRITE(3,*)'*** MANEUVER DETECTION ***'
             CALL REINIT(DT,ZY,ZYM1,LPKKM1,XKKM1,PKKM1)
             E1M1=0.0
             E1M2=0.0
             GOTO 204
        ENDIF

        TIMEM1=TIME
        DATE1=DATE

        ZYM1=ZY
        GOTO 810
        WRITE(6,*)TOTTIM,XKK(1,1),XKK(2,1),PKK(1,1)
C THIS IS WHERE THE SMOOTHING ALGORITHM STARTS
C FIXED INTERVAL SMOOTHING
800     WRITE(*,*) 'SMOOTHING FILTERED DATA WITH A'
        WRITE(*,*) 'FIXED INTERVAL SMOOTHING ALGORITHM'
```

```
      WRITE(*,*) '***=====***'

C     WRITE (*,*) DT,NP,WINDD
      DO 1000 KK=1,NP-1
C     CALL REINIT(DT,ZY,ZYM1,LPKKM1,XKKM1,PKKM1)
       K=NP-KK
       DT=DTS(K+1)

       TIME=TIMEM1-DT
       TOTTIM=TOTTIM-DT
       CALL FINDPHI(PHI,DT)

       DO 901 I=1,2
        XSS(I,1)=XKKS(I,1,K)
901    CONTINUE

       DO 902 I=1,2
        DO 902 J=1,2
         PSS(I,J)=PKKS(I,J,K)
902    CONTINUE

C CALCULATE THE PREDICTED STATE AND ERROR COVARIANCE MATRICES
C    X(K+1|K)=PHI*X(K|K)
       CALL MATMUL (PHI,XSS,2,2,1,XKKM1S)
C    P(K+1|K)=PHI*P(K|K)*PHIT+Q
       CALL MATRAN (PHI,PHIT,2,2)
       CALL MATMUL(PHI,PSS,2,2,2,TEMP6)
       CALL MATMUL(TEMP6,PHIT,2,2,2,TEMP7)
       CALL GETQ(Q,DT)
       CALL MATADD(TEMP7,Q,2,2,1,PKKM1S)


C CALCULATE THE SMOOTHING FILTER GAIN MATRIX
C    AK=P(K|K)*PHIT*INV°P(K+1|K)
       CALL MATINV (PKKM1S,2,IPKKM1S)
       CALL MATMUL (PKKM1S,IPKKM1S,2,2,2,II)
       CALL MATMUL (PSS,PHIT,2,2,2,TEMP1S)
       CALL MATMUL (TEMP1S,IPKKM1S,2,2,2,AK)

       DO 904 I=1,2
         XNNM1(I,1)=XKKS(I,1,K+1)
904    CONTINUE

C CALCULATE THE SMOOTHED STATE ESTIMATE
C    XKKS=X(K|K)+AK*(X(K+1|N)-X(K+1|K)
       CALL MATSUB (XNNM1,XKKM1S,2,1,TEMP2S)
       CALL MATMUL (AK,TEMP2S,2,2,1,TEMP3S)
       CALL MATADD (XSS,TEMP3S,2,1,K,XKKS)

       DO 906 I=1,2
        DO 906 J=1,2
          PNNM1(I,J)=PKKS(I,J,K+1)
906    CONTINUE

C CALCULATE THE SMOOTHED COVARIANCE MATRIX
C    PKKS=P(K|K)+AK*[P(K+1|N)-P(K+1|K)]*AKT
```

```
         CALL MATSUB (PNNM1,PKKM1S,2,2,TEMP4S)
         CALL MATRAN (AK,AKT,2,2)
         CALL MATMUL (AK,TEMP4S,2,2,2,TEMP5S)
         CALL MATMUL (TEMP5S,AKT,2,2,2,TEMP6S)
         CALL MATADD (PSS,TEMP6S,2,2,K,PKKS)

        WRITE(3,*) 'SMOOTHED DATA FOR DATA POINT',K
        WRITE(3,*) 'TIME    VEL.   ACCEL.   HEADING  SPEED'
        WRITE(3,*)TOTTIM,XKKS(1,1,K),XKKS(2,1,K)
        WRITE(6,*)TOTTIM,XKKS(1,1,K),XKKS(2,1,K),PKKS(1,1,K)
1010    FORMAT(1X,5F10.3)
1020    FORMAT(1X,F6.2,3X,F10.1,2X,F11.1,3X,F8.1,3X,F8.1)
1030    FORMAT(1X,F6.2,3(F8.1,2X))

        TIMEM1=TIME
1000    CONTINUE

1100    CONTINUE
1110    FORMAT(I4,2F8.1)
1120    FORMAT(I4,3(F8.1,2X))

        CLOSE(UNIT=2)
        CLOSE(UNIT=3)
        CLOSE(UNIT=4)
        CLOSE(UNIT=5)
        CLOSE(UNIT=6)
        CLOSE(UNIT=9)
        CLOSE(UNIT=8)
        WRITE(*,*) 'FILTERED & SMOOTHED OUTPUT DATA IS LOCATED IN THE'
        WRITE(*,*) 'DATA FILE OUTDATA.DAT.  FOR GRAPHIC RESULTS,'
        WRITE(*,*) 'ENSURE OBSDATA.DAT, FILDATA.DAT, & SMDATA.DAT ARE'
        WRITE(*,*) 'IN THE MATLAB SUB-DIRECTORY AND RUN THE MATLAB'
        WRITE(*,*) 'M-FILE STORM2.M'
        STOP
        END


C*********************************************************************
C                            SUBROUTINES
C*********************************************************************


        SUBROUTINE FINDPHI(PHI,DT)
C ***********************************************************
C       COMPUTES THE VALUES OF THE PHI MATRIX
C ***********************************************************
        REAL*4 PHI(2,2),DT

C       DO 1501 I=3,4
C       DO 1501 J=1,4
                PHI(I,J)=0.0
C501    CONTINUE

C COMPUTE PHI MATRIX
        DO 1500 I=1,2
```

```
         PHI(I,I)=1.0
1500     CONTINUE
         PHI(1,2)=DT
         PHI(2,1)=0.0
C        PHI(2,3)=0.0
C        PHI(2,4)=0.0
C        PHI(1,3)=0.0
C        PHI(1,4)=0.0

         RETURN

         END


         SUBROUTINE INIT(WINDD,XKK,PKK)
C *******************************************************
C        THIS ROUTINE INITIALIZES THE STATE
C        AND ERROR COVARIANCE ESTIMATES
C *******************************************************
         REAL*4 XKK(1,1),PKK(2,2)
         REAL*4 WIND,WINDD

C INITIAL STATE ESTIMATE
         XKK(1,1)=WINDD
         WRITE(*,*) XKK(1,1)
C        XKK(3,1)=0.0
C        XKK(4,1)=0.0

C INITIAL ERROR COVARIANCE ESTIMATE
         PKK(1,1)=1000000.
         PKK(1,2)=0.0
C        PKK(1,3)=0.0
C        PKK(1,4)=0.0
         PKK(2,1)=0.0
         PKK(2,2)=0.25
C        PKK(2,3)=0.0
C        PKK(2,4)=0.0
C        PKK(3,1)=0.0
C        PKK(3,2)=0.0
C        PKK(3,3)=0.0
C        PKK(3,4)=0.0
C        PKK(4,1)=0.0
C        PKK(4,2)=0.0
C        PKK(4,3)=0.0
C        PKK(4,4)=0.0

                 RETURN

                 END

         SUBROUTINE GETQ(Q,DT)
C***********************************************************************
C        ROUTINE TO GET Q MATRIX
C***********************************************************************
         REAL*4 Q(2,2),DT
```

```
C       DO 100 I=1,4
C        DO 100 J=3,4
C00     Q(I,J)=0.0
        Q(1,1)=(DT**4)/4.
        Q(1,2)=(DT**3)/2.
        Q(2,1)=(DT**3)/2.
        Q(2,2)=(DT**2)
C       DO 200 I=3,4
C         DO 200 J=1,4
C00     Q(I,J)=0.0

        RETURN

        END


        SUBROUTINE REINIT(DT,ZY,ZYM1,LPKKM1,XKKM1,PKKM1)
C ****************************************************************************
C       THIS ROUTINE RE-INITIALIZES THE STATE AND ERROR
C       COVARIANCE ESTIMATES
C ****************************************************************************
        REAL*4 DT,XKKM1(2,1),PKKM1(2,2)
        REAL*4 ZX,ZY,ZXM1,ZYM1,LPKKM1(2,2)

C       XDIFF=ZX-ZXM1
C       YDIFF=ZY-ZYM1

C       XKKM1(1,1)=ZX
        XKKM1(1,1)=ZY
C       XKKM1(3,1)=0.0
C       XKKM1(4,1)=0.0

        WRITE(*,*)'REINITIALIZED STATES ARE:'
        DO 100 I=1,2
                WRITE(*,*)XKKM1(I,1)
100     CONTINUE

        PKKM1(1,1)=2.25*LPKKM1(1,1)
        PKKM1(1,2)=0.0
C       PKKM1(1,3)=2.25*LPKKM1(1,3)
C       PKKM1(1,4)=0.0
        PKKM1(2,1)=0.0
        PKKM1(2,2)=0.1111
C       PKKM1(2,3)=0.0
C       PKKM1(2,4)=0.0
C       PKKM1(3,1)=2.25*LPKKM1(3,1)
C       PKKM1(3,2)=0.0
C       PKKM1(3,3)=2.25*LPKKM1(3,3)
C       PKKM1(3,4)=0.0
C       PKKM1(4,1)=0.0
C       PKKM1(4,2)=0.0
C       PKKM1(4,3)=0.0
C       PKKM1(4,4)=0.1111

                RETURN
```

```
                END

        SUBROUTINE MP(XS1,YS1,XS2,YS2,BRG1,BRG2,ZX,ZY)
C ***************************************************************
C       THIS ROUTINE COMPUTES THE ESTIMATED
C        X,Y POSITION OBTAINED FROM MEASUREMENTS
C ***************************************************************
        REAL*4 ZX,ZY
        REAL*4 XS1,YS1,XS2,YS2,BRG1,BRG2
        REAL*4 NUMER,DENOM

C INITIAL STATE ESTIMATE

        NUMER=(-YS2*TAN(BRG2))+(YS1*TAN(BRG1))+XS2-XS1
        DENOM=TAN(BRG1)-TAN(BRG2)

        ZY=NUMER/DENOM
        ZX=(ZY-YS1)*TAN(BRG1)+XS1

        RETURN

        END


        SUBROUTINE ELLIP(XT,YT,P1,P3,P13)
C ***********************************************************
C       THIS SUBROUTINE COMPUTES ERROR ELLIPSE DATA
C       FROM ERROR COVARIANCE DATA
C ***********************************************************
C       DIMENSIONS AND DECLARATIONS
        REAL*4 XT,YT,XP(21),YP(21),A,B,THE1,SIG2X,SIG2Y
        REAL*4 SX,SY,PT,CT,ST,P1,P13,P3

        A=2*P13
        B=P1-P3
        THE1=0.5*ATAN2(A,B)
        A=(P1+P3)/2
        B=0.0
        IF (P13.EQ.0.0) GOTO 10
        B=P13/SIN(2.0*THE1)
10      SIG2X=ABS(A+B)
        SIG2Y=ABS(A-B)
        SX=SIG2X**0.5
        SY=SIG2Y**0.5
        PT=3.141592654/10
        CT=COS(THE1)
        ST=SIN(THE1)

        DO 100 IE=1,21
                XP(IE)=SX*COS(PT*IE)*CT-SY*SIN(PT*IE)*ST+XT
                YP(IE)=SX*COS(PT*IE)*ST+SY*SIN(PT*IE)*CT+YT
                WRITE(7,*)XP(IE),CHAR(9),YP(IE)
100     CONTINUE

        RETURN
```

```
      END

      SUBROUTINE MATMUL(A,B,L,M,N,C)
C ***************************************************************
C     THIS ROUTINE MULTIPLIES TWO MATRICES TOGETHER
C      ° C(L,N) = A(L,M) * B(M,N)
C ***************************************************************
C     DIMENSIONS AND DECLARATIONS
      REAL*4 A(L,M),B(M,N),C(L,N)

      DO 10 I=1,L
      DO 10 J=1,N
       C(I,J)=0.0
10    CONTINUE

      DO 100 I= 1,L
      DO 100 J= 1,N
      DO 100 K= 1,M
       C(I,J) = C(I,J) + A(I,K)*B(K,J)
100   CONTINUE

      RETURN

      END

      SUBROUTINE MATRAN(A,B,N,M)
C ***************************************************
C     THIS ROUTINE TRANSPOSES A MATRIX
C             ° B(M,N) = A'(N,M)
C ***************************************************
C     DIMENSIONS AND DECLARATIONS
      REAL*4 A(N,M), B(M,N)

      DO 100 I= 1,N
      DO 100 J= 1,M
       B(J,I) = A(I,J)
100   CONTINUE

      RETURN

      END

      SUBROUTINE MATSCL(Q,A,N,M,C)
C ****************************************************************
C     THIS ROUTINE MULTIPLIES A MATRIX WITH A SCALAR
C      ° C(N,M) = Q * A(N,M)
C ****************************************************************
C     DIMENSIONS AND DECLARATIONS
            REAL*4 A(N,M), C(N,M), Q

      DO 100 I = 1,N
      DO 100 J = 1,M
```

```
         C(I,J) = Q*A(I,J)
100     CONTINUE

        RETURN

        END


        SUBROUTINE MATSUB(A,B,N,M,C)
C ***************************************************
C       THIS ROUTINE SUBTRACTS TWO MATRICES
C       ° C(N,M) = A(N,M) - B(N,M)
C ***************************************************
C       DIMENSIONS AND DECLARATIONS
        REAL*4  A(N,M),B(N,M),C(N,M)

        DO 100 I = 1,N
        DO 100 J = 1,M
         C(I,J)=A(I,J)-B(I,J)
100     CONTINUE

        RETURN

        END

        SUBROUTINE MATADD(A,B,N,M,L,C)
C***************************************************
C       THIS ROUTINE ADDS TWO MATRICES
C       ° C(N,M) = A(N,M) + B(N,M)
C***************************************************
C       DIMENSIONS AND DECLARATIONS
        REAL*4  A(N,M),B(N,M),C(N,M,L)
        DO 100 I = 1,N
        DO 100 J = 1,M
         C(I,J,L)=A(I,J)+B(I,J)
100     CONTINUE

        RETURN
        END


        SUBROUTINE MATINV (A,N,C)
C***************************************************
C       THIS ROUTINE COMPUTES THE INVERSE OF
C       A MATRIX
C          C(N,N)=INV [A(N,N)]
C***************************************************
C        DIMENSIONS AND DECLARATIONS
         REAL*4 A(N,N),C(N,N),D(20,20)
         DO 100 I = 1,N
          DO 100 J = 1,N
100           D(I,J)=A(I,J)

         DO 115 I=1,N
          DO 115 J=N+1,2*N
115      D(I,J)=0.0
```

```
      DO 120 I=1,N
       J=I+N
120   D(I,J)=1.0

      DO 240 K=1,N
       M=K+1
       IF (K.EQ.N) GOTO 180
       L=K
       DO 140 I=M,N
140    IF (ABS(D(I,K)).GT.ABS(D(L,K))) L=I
       IF (L.EQ.K) GOTO 180

       DO 160 J=K,2*N
        TEMP=D(K,J)
        D(K,J)=D(L,J)
160    D(L,J)=TEMP

180    DO 185 J=M,2*N
185    D(K,J)=D(K,J)/D(K,K)

       IF (K.EQ.1) GOTO 220
       M1=K-1
       DO 200 I=1,M1
        DO 200 J=M,2*N
200   D(I,J)=D(I,J)-D(I,K)*D(K,J)

      IF (K.EQ.N) GOTO 260

220   DO 240 I=M,N
       DO 240 J=M,2*N
240      D(I,J)=D(I,J)-D(I,K)*D(K,J)

260   DO 265 I=1,N
       DO 265 J=1,N
        K=J+N
265   C(I,J)=D(I,K)

      RETURN
      END
```

LIST OF REFERENCES

1. Anthes, R. A., *Tropical Cyclones*, pp. 1-10, Boston, MA, 1982.

2. Galinis W. J., *Fixed Interval Smoothing Algorithm for an Extended Kalman Filter for Over the Horizon Ship Tracking*, M.S.E.E. Thesis, Naval Postgraduate School, Monterey, CA, March 1989.

3. Gelb, Arthur, *Applied Optimal Estimation*, pp. 100-128, MIT Press, Cambridge, MA, 1974.

4. Lewis, F. L., *Optimal Estimation with an Introduction to Stochastic Control Theory*, pp. 127-128, School of Electrical Engineering, Georgia Institute of Technology, Atlanta, GA, 1986.

5. Meditch, J. S., *Stochastic Optimal Linear Estimation and Control*, pp. 204-205, McGraw-Hill Book Co., New York, NY, 1969.

6. Joint Typhoon Warning Center, Guam, Mariana Islands, *1985 Annual Tropical Cyclone Report*, pp. 64-77, Naval Oceanography Command Center/Joint Typhoon Warning Center, 1985.

7. Ahrens, C. D., *Meteorology Today*, pp. 426-428, San Francisco, CA, 1985.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor H. A. Titus, Code 62Ts Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	2
5.	Professor R.Cristi, Code 62Cr Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Genel Kurmay Baskanligi Personel Dairesi Bakanliklar - Ankara / TURKEY	2
7.	AR-GE Bakanliklar - Ankara / TURKEY	2
8.	Deniz Kuvvetleri Komutanligi Personel Daire Baskanligi Bakanliklar-Ankara / TURKEY	1
9.	Kara Harp Okulu Komutanligi Okul Kutuphanesi ve Elektrik Bolumu Kutuphanesi Bakanliklar - Ankara / TURKEY	2
10.	Deniz Harp Okulu Komutanligi Okul Kutuphanesi ve Elektrik Bolumu Kutuphanesi Tuzla - Istanbul / TURKEY	2

11. Hava Harp Okulu Komutanligi
Okul Kutuphanesi ve
Elektrik Bolumu Kutuphanesi
Istanbul / TURKEY 2

12. Fakulte ve Yuksek Okullar Komutanligi
Kutuphanesi
Dikimevi - Ankara / TURKEY 1

13. Fakulte ve Yuksek Okullar Komutanligi
Kutuphanesi
Cankurtaran - Istanbul / TURKEY 1

14. Gurkan TURKES
NPGS. SMC.# 2983
Monterey, CA 93943 1

15. Asim MUTAF
Esendere Mah. 52/29 sk. No.:14/2
Guzelyali-Izmir / TURKEY 2

16. Bogazici Universitesi
Elektrik Fakultesi
Istanbul / TURKEY 1

17. Ortadogu Universitesi
Elektrik Fakultesi
Ankara / TURKEY 1

18. Istanbul Teknik Universitesi
Elektrik Fakultesi
Istanbul / TURKEY 1

19. Cinarli Teknik Lisesi
Elektrik Bolumu
Cinarli-Izmir/ TURKEY 1